Volume 16, No. 1
February 2018
ISSN: 1545-679X

INFORMATION SYSTEMS EDUCATION JOURNAL

In this issue:

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (http://isedj.org). Our sister publication, the Proceedings of EDSIGCON (http://www.edsigcon.org) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 30%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of AITP-EDSIG who perform the editorial and review processes for ISEDJ.

2018 AITP Education Special Interest Group (EDSIG) Board of Directors

INFORMATION SYSTEMS
EDUCATION JOURNAL

Connecting the Dots and Nodes: A Survey of Skills Requested by Employers for Network Administrators

Gerard Morris
morrisgj@ msudenver.edu

Janos Fustos
fustos@ msudenver.edu

Wayne Haga
haga@ msudenver.edu

Computer Information Systems and Business Analytics Department
Metropolitan State University of Denver
Denver, CO 80202, USA

Abstract

One definition of a network administrator describes a person who works with computer infrastructures with an emphasis on networking. To determine the specific skills required of a network administrator by employers, data was collected from 698 nationwide job advertisements on Dice.com. The data collection focused on technical skills rather than soft skills. The requested skills were then broken into various categories in the networking and computing world such as routing protocols, operating systems, virtualization technologies, etc. Educational requirements in terms of degrees and certificates were also tabulated. The results show a great breadth in the requested skills and the summary results will be helpful for curriculum development/review, for career advising, and to faculty teaching in the Information Systems field.

Keywords: network administrator, skills, networking, job description, education.

1. INTRODUCTION

The Role of Computer Networking

Technology continues to increasingly become an integral part of our lives. Networking is an important, critical aspect of technology. We use it for almost everything: work, education, entertainment, shopping, keeping in touch with loved ones, meeting new people. A computer network allows nodes to share resources. They allow for sharing of files, sharing of printers, communications such as email and chat, remote access, and data protection.

There are numerous applications in different fields of business. In marketing they are used to collect, exchange, and analyze data to target customers. In sales, teleshopping allows customers to quickly search for products, read reviews, compare prices and process an order. Computer networks are now used extensively in manufacturing, including computer assisted manufacturing (CAM) and computer assisted design (CAD), allowing several employees to work on projects simultaneously. Financial services are almost completely dependent on networks - allowing transfer of funds instantaneously, foreign exchanges, stock purchase and sales, and many other applications. Teleconferences and videoconferencing allow easy communication in meetings with worldwide participants. The entertainment industries can now provide video on demand of almost unlimited content. Other common applications include information services, electronic messaging, electronic data interchange, directory services, and cellular telephone (Tiwari, 2016).

The Role of a Network Administrator

According to Wikipedia, "A **network administrator** maintains computer infrastructures with emphasis on networking. Responsibilities may vary between organizations, but on-site servers, software-network interactions as well as network integrity/resilience are the key areas of focus" (en.wikipedia.org, 2012).

Several key duties of a network administrator include: working with users to establish network specifications, evaluating network performance issues, maintaining network performance, securing the network, preparing and supporting users, continuously upgrading the network, meeting financial requirements/budgeting, continually updating job knowledge, protecting the organization's confidential information, and aiding in accomplishing the organization goals (Hiring.monster.com, 2004).

Job Demand and Wages

The 2015 edition of the U.S. Bureau of Labor Statistics Occupational Handbook predicts that employment for network and computer systems administrators will grow 8% between 2014 and 2024, which is about the average growth for all occupations. This will increase the number of jobs in 2014 (382,600) by 30,200 to 412,800 in 2024. They report a median annual wage as of May 2016 to be $79,700. This was slightly lower than the median of all computer occupations ($82,860), but much higher than the median of all occupations ($37,040). The lowest 10% in the field earned less than $48,870 and the highest 10% earned more than $127,610.

The largest employers were computer systems design (16%), information (11%), state, local, and private educational services (10%), finance and insurance (8%), and administrative and support services (7%). Of these categories, employees working in Information earned the highest (median $85,960), and not surprisingly employees in education services earned the least (median $68,510) (US DoL, 2015).

Roberthalf.com paints a rosier salary picture, and projects a starting salary in 2017 for network administrators ranging from $78,000 to $117,750 - a 4% increase over 2016. Salaries vary widely by location (Badion, 2016).

Salary.com breaks Network Administrator salaries into 5 main categories, Network Administrator I (median $57,601), Network Administrator II (median $75,108), Network Administrator III (median $87,324), Network Administrator IV (median $103,979), and Network Administrator V (median $122,919). This site also allows prospective employees to check median salaries based on job title, education, years experience, location, and other factors (Salary.com, 2017).

2. EDUCATION

A search of over 7500 college/universities on mymajors.com for programs titled "Network and System Administration/Administrator found a total of 246 program offerings. Eighty of these were 4-year programs, 151 were 2-year programs, and 15 were less than 2-year programs. One hundred and thirty-three of the 246 were offered through public institutions, 11 were private not-for-profit institutions, and 102 were private for-profit institutions. A majority (186 out of 246) of them were offered as online programs. Closely related program searches revealed 874 institutions with "Computer Systems and Networking" programs and 292 institutions with "System, Networking, and LAN/WAN Management" programs. Of course there are many additional institutions that may offer a concentration, emphasis, or certificate under a more general Information Systems, Computer Science, Information Technology, or Engineering degree program (mymajors.com, 2017).

Professional organizations have offered recommendations of how much networking and telecommunications topics should be included in general bachelor computer degree programs. The "Information Technology 2008 Curriculum guidelines for Undergraduate Degree Programs in Information Technology" recommends 22 core hours in networking in an Information Technology degree. An additional 59 hours are recommended in closely related areas (platform technologies, information assurance and security, and web systems and technologies. This joint work by the Association of Computing Machinery (ACM) and Institute of Electrical and Electronics Engineers (IEEE) has not been updated since 2008 (ACM, 2008). Updated guidelines are expected to be completed in 2017. The "IS 2010 Curriculum Guidelines for Undergraduate Degree Programs in Information Systems" do not specify any recommended number of hours in networking and telecommunications but instead just list Enterprise Architecture and IT Infrastructure as two of the seven recommended areas of study (ACM, 2010). The Accreditation Board for Engineering and Technology (ABET) also does not indicate the number of hours in different areas, just that students are required to be exposed to networking topics (ABET, 2010).

3. WHAT EMPLOYERS WANT

Job ads for computer specialists vary greatly in listed skill requirements. Kasson notes "Many employers required skill sets seem to include everything but the ability to teleport and build a Shaker barn; the lengthy requisites of skills and experience seem achievable only by candidates who've spent the past four decades using a hundred different programming languages and platforms to excel at fifty different, complicated jobs." Other job ads may just list the job title and a few very general skills such as problem solving. Sometimes an employer is unsure what it wants/needs in a new hire, and will add additional responsibilities to be "safe". At some companies, HR managers write the job descriptions and they are written very generic and vague and may not represent at all what is really needed (Kasson, 2015).

A review of the literature reveals several efforts to identify employer needs in IS/IT. Janicki developed a survey with the assistance of 20 IT professionals from various areas to analyze employer needs. Employers were asked to rate desired knowledge in different topics from 1 (not needed) to 4 (expert level needed). From 308 responses to the survey they summarized results by curriculum topic. They conclude that "regardless of the occupation, the employers expected a working knowledge of system documentation, security, IT ethics and privacy, problem identification, specific programming language and process analysis (Janicki, Lenox, Logan, & Woratschek, 2008). Nelson presents a methodology for using Latent Semantic Clustering to analyze a comparison of job skills requirements between small and large firms (Nelson, Ahmad, Martin, & Litecky, 2007). Morris identifies and categorizes skills requested for Network Engineers (Morris, Fustos, & Haga, 2012).

Shankararaman and Gottipati present a solution model for generating a skills report based on the Skills Framework for the Information Age (SFIA) (Shankararaman & Gottipati, 2016). The skill generator takes in curriculum and LinkedIn profile data as input, and generates a list of recommended jobs that match the student's skills. Legier presents an analysis of job characteristics of information systems graduates based on 72 graduates of an information systems program at a midwestern university (Legier, Woodward, & Martin, 2013). Litecky uses a web mining application to extract nearly a quarter of a million unique IS/IT job ads from Monster.com, HotJobs.com, and SimplyHired.com (Litecky, Aken, Ahmed, & Nelson, 2010). The data was then analyzed using cluster analysis. Mirza uses Holland codes to try to identify suitable job roles for IT job seekers by analyzing their personality (Mirza, Mulla, Parekh, Sawant, & Singh, 2015).

A review of the literature revealed no papers specifically addressing the technical skills employers are looking for in a Network Administrator position. This research is intended to fill that gap.

4. METHODOLOGY

A national search for jobs with "Network Administrator" in the title was performed at Dice.com. Dice.com is one of the major job boards for technical positions. A total of 698 sequential job positions were downloaded from September 2014 through October 2016. Obvious duplicate jobs were eliminated. Only positions with the exact title "Network Administrator" were selected. Job requirements for the 698 jobs were examined. The requested skills were categorized by types of protocols, standards, operating systems, etc. and tabulated. This paper addresses just the technical skills employers are asking for. As one would expect, many of the ads also requested generic and "soft skills".

5. RESULTS

The following tables show how the results were categorized. The tables show routing protocols, other protocols, LAN and WAN topics, operating systems and types of servers plus server technologies, security protocols, languages and scripting, education requirements such as degree and certificate requirements, and networking vendors.

Table 1 indicates the number of jobs that listed specific routing protocols or the term "Router". Border Gateway Protocol (BGP) was the most frequently requested protocol appearing in 14.2% of the positions. It is an exterior gateway protocol

(EGP) that is used to link autonomous systems. Both Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP) are interior gateway protocols (IGP) and occur in 10.9% and 8.9% respectively of the jobs. The general term "Router" in the job ads just adds emphasis to the importance of this area in general. Cisco's Internetwork Operating System (IOS) is used to configure Cisco switches and routers and was requested by 9.6% of companies. The Juniper switch and router configuration language, Junos OS, also appears in Table 1. Three entries in the table are First Hop Routing Redundancy (FHRP) protocols: Hot Standby Router Protocol (HSRP), Virtual Router Redundancy Protocol (VRRP), and Gateway Load Balancing Protocol (GLBP).

Routing Protocols/Concepts	n	%
Router	365	52.3%
Border Gateway Protocol (BGP)	99	14.2%
Open Shortest Path First (OSPF)	76	10.9%
Cisco Internetwork Operating System (IOS)	67	9.6%
Enhanced Interior Gateway Routing Protocol (EIGRP)	62	8.9%
Hot Standby Router Protocol (HSRP)	26	3.7%
Virtual Router Redundancy Protocol (VRRP)	12	1.7%
Junos OS	11	1.6%
Gateway Load Balancing Protocol (GLBP)	5	0.7%

Table 1. Routing-related Protocols/Concepts

Table 2 lists the other protocols requested. As one would expect, Internet Protocol (IP) had the largest percentage of requests (32.2%), due to the vast body of knowledge it covers such as subnetting, and understanding addressing in terms of route aggregation. IPv4 and IPv6, the specific versions of IP, showed up in only 2.6% and 1.7% respectively of the jobs. Domain Name System (DNS) is the second most requested knowledge area as expected as it is an integral part of networking. In terms of transport-layer protocols, Transmission Control Protocol (TCP) was requested in 22.8% of the ads but User Datagram Protocol (UDP) only showed up in 1.1% of ads. Voice over IP (VoIP) is the fifth most requested item (22.1%) indicating the increasing role of VoIP systems in business.

Other Protocols	n	%
Internet Protocol (IP)	225	32.2%
Domain Name System (DNS)	191	27.4%
Hypertext Transfer Protocol (HTTP)	184	26.4%
Transmission Control Protocol (TCP)	159	22.8%
Voice over Internet Protocol (VoIP)	154	22.1%
Dynamic Host Configuration Protocol (DHCP)	138	19.8%
Simple Mail Transfer Protocol (SMTP)	37	5.3%
Simple Network Management Protocol (SNMP)	93	6.2%
File Transfer Protocol (FTP)	26	3.7%
Windows Internet Name Service (WINS)	21	3.0%
Network Address Translation (NAT)	19	2.7%
Secure Hypertext Transfer Protocol (HTTPs)	19	2.7%
IPv4	18	2.6%
IPv6	12	1.7%
User Datagram Protocol (UDP)	8	1.1%
Internet Message Access Protocol (IMAP)	8	1.1%
Secure File Transfer Protocol (sFTP)	7	1.0%

Table 2. Other Protocols

The most requested application-oriented protocol is HTTP (26.4%). In addition to DNS, DHCP was also a frequently requested networking protocol.

WAN Services	n	%
WAN	273	39.1%
Multiprotocol Label Switching (MPLS)	68	9.7%
T1	18	2.6%
Frame Relay	11	1.6%
T3/DS-3	8	1.1%
Digital Subscriber Line (DSL)	8	1.1%
Synchronous Optical Networking (SONET)	6	0.9%

Table 3. WAN Services

Table 3 shows the most-requested Wide Area Network (WAN) services. General wide area network experience was requested in 39.1% of the positions. Multiprotocol Label Switching

(MPLS) was the most requested. It was a surprise to see T1s requested at all given their low bandwidth and also unexpected to see it requested more than T3s.

LAN topics are tabulated in Table 4. Knowledge of Virtual Local Area Networks (VLAN) was the most requested topic (8.2%) in this section with Spanning Tree Protocol - 802.1d the next most requested (3%).

LAN Topics	n	%
Virtual Local Area Network (VLAN)	57	8.2%
Wireless LAN (WLAN)	45	6.4%
Spanning Tree Protocol - 802.1d	21	3.0%
Aruba Wireless	10	1.4%
VLAN Trunking Protocol (VTP)	9	1.3%
Network Time Protocol (NTP)	7	1.0%

Table 4. LAN Topics

Operating Systems and Related Areas	n	%
Active Directory	226	32.4%
Windows Server 2000/2003/2008/2012	166	23.8%
Linux	134	19.2%
Windows 7	58	8.3%
Group Policy	57	8.2%
Unix	50	7.2%
Windows XP	31	4.4%
Red Hat	27	3.9%
LDAP	16	2.3%
CentOS	12	1.7%
Mac OS X	12	1.7%
Android	12	1.7%
Domain Controllers	12	1.7%
Windows 10	5	0.7%
Debian	5	0.7%

Table 5. Operating Systems and Related Areas

Operating systems and related concepts are tabulated in Table 5. Some version of Windows Server was requested in 23.8% of positions. Areas related to Windows Server such as Active Directory (32.4%), Group Policy (8.2%) and Lightweight Directory Access Protocol (2.3%), and Domain Controllers (1.7%) are also featured in the table.

Linux along with its variations Red Hat, CentOS, and Debian also scores high. Of the Windows client operating systems the most requested was Windows 7 at 8.3%.

In terms of Database Managements Systems (DBMS) knowledge required, the term Structured Query Language (SQL) appeared in 14.9% of positions – see Table 6. The two most requested DBMSs are Microsoft SQL Server (8.3%) and Oracle (2.7%). The higher percentage for SQL Server could correlate with the large number of positions using a version of Windows Server for their network operating system.

Database-related Solutions	n	%
SQL	104	14.9%
Microsoft SQL Server	58	8.3%
Oracle	19	2.7%
MySQL	12	1.7%

Table 6. Database-related Solutions

The requested web servers are tabulated in Table 7. Microsoft IIS Server (6.4%) is the most requested. The other requested system is Apache, which runs on Linux machines.

Web Servers	n	%
Microsoft IIS Server	45	6.4%
Web Server	20	2.9%
Apache	10	1.4%

Table 7. Web Servers

Other Servers	n	%
Microsoft Exchange	200	28.7%
SharePoint	51	7.3%
File Server	28	4.0%
Print Server	8	1.1%
Application	4	0.6%

Table 8. Other Servers

In addition to the aforementioned database and web servers, other servers such as email, file and print servers also appeared in the ads. The most requested of these by far is Microsoft Exchange Server (28.7%) – see Table 8.

Table 9 shows the popularity of virtualization, a technology that is often associated with cloud computing. The most requested virtualization software is VMWare in 31.5% of the jobs with Microsoft's Hyper-V a distant second (7.7%).

Server, Storage, and Virtualization Technologies	n	%
VMware	220	31.5%
Storage Area Network (SAN)	136	19.5%
Virtualization	97	13.9%
Hyper-V	54	7.7%
VMWare Vsphere	36	5.2%
ESX	35	5.0%
Network Attached Storage (NAS)	32	4.6%
High Availability	27	3.9%
RAID	7	1.0%
Blade/Blade Servers	6	0.9%

Table 9. Server, Storage, and Virtualization Technologies

Security Protocols/Technologies	n	%
Virtual Private Networks (VPN)	228	32.7%
Firewall	169	24.2%
Anti-virus	77	11.0%
Secure Socket Layer (SSL)	44	6.3%
Internet Protocol Security (IPsec)	43	6.2%
SonicWall	31	4.4%
Intrusion Detection System (IDS)	22	3.2%
Dynamic Multipoint Virtual Private Network (DMVPN)	12	1.7%
Demilitarized Zone (DMZ)	9	1.3%

Table 10. Security Protocols/Technologies

The importance of security protocols and technologies for the network administrator position is illustrated in Table 10. General concepts like Virtual Private Networks (VPN) and firewalls score high. The protocols Secure Socket Layer (SSL) and Internet Protocol Security (IPsec) were also requested in a reasonable number of jobs, at 6.3% and 6.2% respectively.

Scripting in general is requested in 10% of ads (Table 11) while C (5.2%) and Microsoft's scripting platform, PowerShell (3.4%), were the two most requested tools.

Languages/Scripting	n	%
Scripting	70	10.0%
C	36	5.2%
PowerShell	24	3.4%
Perl	18	2.6%
PHP	11	1.6%
Python	10	1.4%
C#	4	0.6%

Table 11. Languages/Scripting

The next two tables show degree and certificate requirements for the 698 jobs surveyed. Table 12 shows the degrees requested. A Bachelors degree was the most commonly requested with zero requests for a Masters degree. A small percentage required only a High School diploma. Degrees in Computer Science and Information Systems were the two most common computer-related degrees for a network administrator position. One could argue that an Engineering degree is the one that is least related to the job title but it scored second in terms of specific degree requests.

Degree Requirements	n	%
Bachelors Degree	287	41.1%
Degree in Computer Science	199	28.5%
Degree in Engineering	119	17.0%
Degree in Information Systems	94	13.5%
High School Diploma	16	2.3%

Table 12. Degree Requirements

The numerous certificates required in the positions are shown in Table 13 and broken down by company/organization/area: Cisco, Microsoft, CompTIA, Security, and Juniper. As expected, the Cisco and Microsoft certificates were the most requested. Cisco Certified Network Associate (CCNA) scored the highest with Cisco Certified Network Professional (CCNP) and Microsoft Certified Systems Engineer (MCSE) in second and third positions respectively.

Certificate Requirements	n	%
Cisco		
Cisco Certified Network Associate (CCNA)	196	28.1%
Cisco Certified Network Professional (CCNP)	109	15.6%
Cisco Certified Internetwork Expert (CCIE)	25	3.6%
Cisco Certified Design Professional (CCDP)	9	1.3%
Cisco Certified Entry Networking Technician (CCENT)	7	1.0%
Microsoft		
Microsoft Certified Systems Engineer (MCSE)	72	10.3%
Microsoft Certified Systems Administrator (MCSA)	25	3.6%
Microsoft Certified IT Professional (MCITP)	18	2.6%
CompTIA		
Network +	39	5.6%
A+	34	4.9%
Security		
Security +	35	5.0%
Certified Information Systems Security Professional (CISSP)	11	1.6%
DoD 8570 Compliant	8	1.1%
Information Technology Infrastructure Library (ITIL)	22	3.2%
Juniper		
Juniper Networks Certified Associate (JCNIA)	7	1.0%
Juniper Networks Certified Professional (JNCIP)	4	0.6%

Table 13. Certificate Requirements

The data in Table 13 is important information, as are the protocols and technologies in the other tables, to share with students.

The last table indicates which vendors appeared in the job requirements. Juniper makes a good showing after the expected numbers 1 and 2, Cisco and Microsoft respectively. It is important to note a vendor's name could show up in an ad in relation to a switch/router, a certificate, or a product. An example of the latter would be Palo Alto, which could occur due to its firewall technology.

Vendors	n	%
Cisco	427	61.2%
Microsoft	277	39.7%
Juniper	70	10.0%
Citrix	60	8.6%
HP	55	7.9%
Dell	54	7.7%
Apple	46	6.6%
SolarWinds	43	6.2%
Palo Alto	37	5.3%
Avaya	25	3.6%
Aruba	20	2.9%
IBM	15	2.1%
Barracuda	14	2.0%
Nortel	12	1.7%
Alcatel-Lucent	3	0.4%

Table 14. Vendors

6. CONCLUSIONS

The results contain a vast store of information for faculty teaching in the networking area and also for curriculum development. The first 13 tables show the breadth of knowledge that could be expected from a person applying for a network administrator position, skills in such diverse areas as routing protocols and switch/router configuration languages (Table 1), operating systems (Table 5), server/storage/virtualization technologies (Tables 6 – 9) and security protocols and technologies (Table 10). The results also give an idea of some of the most important skills and concepts in terms of the requested numbers with IP (32.2%), DNS (27.4%), Active Directory (32.4%), VMWare (31.5%), and VPNs (32.7%) scoring high in the requested skills among the 698 ads.

The tables can be used as a basis to review/develop the networking part of the computing curriculum. Faculty could look at the top two entries or so in each table or at entries that were requested in more than 20% of the job ads and see if these items are covered in the courses. How detailed should the coverage be? An introductory level seems reasonable as students will have to continue learning for life on the job and preparing for certifications. Looking at Table 1 – Routing Protocols/Concepts as an example we see that BGP and OSPF should be covered. The third item in that table is the Cisco IOS. So the concepts of OSPF should be covered and a simple configuration scenario for OSPF could be demonstrated using some IOS simulator package. Another example is Table 3 – WAN Services. Textbooks typically cover T1/T3s, Frame Relay, and SONET, but this table shows it is worthwhile to add some coverage of MPLS.

The level of expertise expected was not addressed directly in the job ads but they did indicate years of experience needed. 115 positions requested a minimum of three or more years of experience. Coupled with the 28.1% of job ads that expected Cisco's CCNA certificate, and the 15.6% that expected Cisco's top CCNP certificate, this indicates that these are not entry level positions in general.

The results also allow faculty to advise students in terms of education and certificate requirements. Degree-seeking students will be happy to learn that 41% of the positions requested a Bachelors degree. On the other hand, the demand for certificates shows how demanding the workplace can be with 28.1% of the positions requesting Cisco's CCNA certificate. This certificate was found to be so challenging that Cisco split it into two exams. Requiring this CCNA certificate in the ads shows the technical level of expertise expected in the routing and switching areas. Further evidence of this is the 15.6% of ads requesting the CCNP, the advanced Cisco networking certificate.

In summary these results can help faculty in teaching, career advising, and curriculum review and development.

7. FUTURE RESEARCH

As noted previously, this paper addresses just the technical skills requested by employers. It may be interesting to try to also measure the frequency/importance of various generic/soft skills requested.

From the ads, we were unable to directly discern the level of mastery of the various skills employers were looking for. This would likely require doing a survey of employers and asking them to rate the requested skill level from 1 to 5, where 1 would be a superficial knowledge and 5 would be complete expertise.

8. REFERENCES

ABET (2010). Criteria for accrediting computing programs 2016-2017. Retrieved May 16, 2017 from http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2016-2017/

ACM (2008). Information Technology 2008 – Curriculum guidelines for undergraduate degree programs in information technology. Retrieved May 16, 2017 from http://www.acm.org//education/curricula/IT2008%20Curriculum.pdf

ACM (2010). IS 2010 - Curriculum guidelines for undergraduate degree programs in information systems. Retrieved May 16, 2017 from https://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf

Badion, C. (2016). Why it's a great time to be a network administrator. Retrieved May 26, 2017 from https://www.roberthalf.com/technology/blog/network-administrator-great-career-rising-salary.

Hiring.monster.com (2004). Network administrator sample job description. Retrieved May 25, 2017 from https://hiring.monster.com/hr/hr-best-practices/recruiting-hiring-advice/job-descriptions/network-administrator-job-description-sample.aspx

Janicki, T., Lenox, T., Logan, R., & Woratschek, C. (2008). Information systems technology employer needs survey: Analysis by curriculum topic. *Information Systems Education Journal*, 6 (18), 3-16.

Kasson, G. (2015). Do employers want too much from candidates? Retrieved May 26 from http://insights.dice.com/2015/03/09/do-employers-want-too-much-from-candidates/

Legier, J., Woodward, B., & Martin, N. (2013). Reassessing the skills required of graduates of an information systems program: an updated analysis. *Information Systems Education Journal*, 11 (3).

Litecky, C., Aken, A., Ahmed, A., & Nelson, J. (2010). Mining for computing jobs. *IEEE Software*, 27(1), 78–85.

Mirza, I., Mulla, S., Parekh, R., Sawant, S., & Singh, K. (2015). Generating personalized job role recommendations for the IT sector through predictive analytics and personality traits. *Proceedings of 2015 International Conference on Technologies for Sustainable Development (ICTSD)*, 4 pages.

Morris, G., Fustos, J., Haga, W. (2012). Preparing for a career as a network engineer. *Information Systems Education Journal*, 10(1), 13-20.

mymajors.com (2017). College Search. Retrieved May 28, 2017 from https://www.mymajors.com/find-a-college/.

Nelson, J., Ahmad, A., Martin, N., & Litecky, C. (2007). A comparative study of IT/IS job skills and job definitions. *SIGMIS-CPR'07*, 168-170.

Salary.com (2017). Network administrator salaries. Retrieved May 25, 2017 from http://www1.salary.com/Network-Administrator-I-Salaries.html.

Shankararaman, V., & Gottipati, S. (2016). Mapping information systems student skills to industry skills framework. *Proceedings of 2016 IEEE Global Engineering Education Conference*, 248-253.

Slayford, S. (2014). Network engineer vs. network administrator. Retrieved May 24, 2017 from http://www.ehow.com/info_8595642_network-engineer-vs-network-administrator.html

Tiwari, R. (2016). What are the goals of establishing computer networks? Retrieved May 25, 2017 from http://mpstudy.com/what-are-the-goals-of-establishing-computer-networks-2/

US DoL (2015). Occupational Outlook Handbook. Retrieved May 16, 2017 from https://www.bls.gov/ooh/computer-and-information-technology/network-and-computer-systems-administrators.htm

Wikipedia.com (2012). Network administrator. Retrieved May 24, 2017 from https://en.wikipedia.org/wiki/Network_administrator

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2017 Meritorious Paper. The acceptance rate is typically 15% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2017.

Introducing the Cloud in an Introductory IT Course

David M. Woods
woodsdm2@miamioh.edu
Computer & Information Technology Department
Miami University Regionals
Hamilton, OH 45011, USA

Abstract

Cloud computing is a rapidly emerging topic, but should it be included in an introductory IT course? The magnitude of cloud computing use, especially cloud infrastructure, along with students' limited knowledge of the topic support adding cloud content to the IT curriculum. There are several arguments that support including cloud computing in an introductory course. In light of this, several cloud computing activities based on the Amazon Web Services (AWS) cloud offerings were added to an introduction to IT course. Student and instructor perceptions of the activities and the AWS service are evaluated and future plans are discussed.

Keywords: Cloud Computing, IT Education

1. INTRODUCTION

Teaching an introductory course in an IT curriculum presents several challenges. There is an abundance of content that supports later classes in the course of study and the course also needs to accommodate students with a with a range of previous experience and knowledge. Additionally, the course needs to have content to explore the wide range of career options for IT professionals and prompt students to start developing longer term plans for making use of the IT knowledge they are acquiring. Finally, with the constant evolution of technology, the course needs continuous updates to include new technologies while avoiding the latest fads that end up being hype without substance.

This paper discusses an effort to update an introductory IT course to add content on cloud computing. Specifically the added content was intended to provide students with experience working with cloud based infrastructure and also show how cloud offerings affect strategic aspects of IT decision making. The AWS Educate program was used to in course assignments and is also evaluated.

Students had little prior knowledge of cloud computing and were surprised to learn how widely cloud infrastructure is used and found that companies they recognize are making significant use of cloud infrastructure. Students reported strong agreement that the cloud computing content and assignments helped them learn about the cloud and should be included and possibly expanded in future offerings of the course. The AWS Educate program was also found to provide good resources to support the cloud computing content in the course.

2. BACKGROUND

Cloud computing is widely discussed, with mainstream media suggesting that "2015 is, by all appearances, the year that the enterprise cloud is turning into a huge market."(Kelleher, 2015). But should it be covered in an introductory IT course? There are many facets to this question – will the students see the value of the material, does it fit with other course content, will it give students knowledge that employer's want, does it fit into the department's curriculum? Many factors should be considered in this decision. To start with, is the cloud going to have a permanent impact on IT, or will it be another passing fad?

Amazon Web Services (AWS) reported $3.66 Billion of revenue in the first quarter of 2017 (Krazit, 2017). Estimates of the number of servers in AWS data centers are in the range of 2.8 million to 5.6 million (Morgan, 2014). Looking at cloud offerings that students may be more familiar with, sixty-one (61) of one hundred (100) top universities are using Google Apps for Education (Mills, 2011). Netflix, a leading video streaming service, has recently completed migrating all of the infrastructure supporting their service to the cloud (Izrailevsky, 2016). These and many more examples indicate that cloud software and infrastructure has become a core part of the IT world

Another factor to consider when adding content to an introductory IT course is whether it will support other courses in the curriculum. Ideally, material in the introductory course will provide an introduction to support more detailed exploration of topics in later courses.

Rehman, et. al. (2015) provide a useful review of cloud computing courses. They find a mix of dedicated courses focusing on how cloud computing works or using a particular cloud based tool and more traditional courses that make use of cloud tools or infrastructure.

In a study comparing the use of cloud-based versus local-based infrastructure to support a web development course, Pike, Pittman, & Hwang (2017) noted that students using cloud-based infrastructure "encountered greater difficulties than anticipated in gaining proficiency with the technology." As they note, this could be addressed by providing a cloud computing introduction in the web development course or in a pre-requisite course.

Lawler (2011) notes that cloud computing connects to many aspects of the IS 2009 Curriculum Model, and presents a model program with significant cloud computing content in the first year of study. Lawler also makes an interesting observation about the impact of cloud computing on the culture of organizations, including internal IT organizations, and proposes addressing change management aspects of cloud computing in the first year of the IS program. Both of these support introducing cloud computing early in the curriculum.

In reviewing our current curriculum, cloud infrastructure or tools could be used in several courses including networking, database, security, web development, and capstone courses and there are existing examples of cloud use in all of these courses. This consideration also argues for including cloud computing content in the introductory IT course.

In considering whether cloud computing fits with the department's curriculum, there were some arguments that it was not a good fit. These were based on the view that cloud computing hid too many details. For example, students using cloud servers would not have to know how to build servers and install operating systems from scratch. There were also arguments that the cloud was too automated. There is value to these arguments, but they must be balanced with the numerous arguments for adding cloud content. To address these concerns, extra attention was given to discussing the strengths and weaknesses of the automation and standardization enabled by cloud technologies.

A final consideration before concluding that cloud content should be added to the course is the possibility that students will have previous knowledge on the topic, which would reduce the necessity for adding the content. Casual discussions with students and observation of the technologies used by students show that they actively use a wide range of tools, apps, and services that use cloud infrastructure. However, the cloud infrastructure is hidden from the end users, so students are not aware that they are making use of cloud computing. As Serapiglia (2013, p. 58) notes "students that I found in my class room were walking in knowing that magic was occurring, and not knowing how." This offered the opportunity for discussions that revealed the "magic" of software and services that students regularly used.

Having concluded that there is value in adding cloud computing content to an introductory IT course, the next questions are what the content should cover and what cloud tools should be used.

Recent efforts have considered a number of different tools including locally run clouds (Hwang, Pike, & Mason, 2016) that would provide a wide range of opportunities for students to design, configure, and manage clouds in addition to using them. However, this may involve significant costs and technical support. For smaller institutions with fewer resources, a public cloud like Microsoft Azure offers another option (Mew, 2016).

At the time these course revisions were being developed, Amazon's AWS Educate (AWS Educate, n.d.) program was relatively new. AWS Educate offers access to a wide range of cloud infrastructure and provides small grants of AWS resources to instructors and students to support these. The opportunity for the instructor to further explore the AWS Educate program combined with free resources for students made this a reasonable choice. An added benefit was that the university's LMS used AWS infrastructure allowing discussion of how this benefitted students.

In considering the content to add, the previously discussed idea of supporting other courses in the curriculum was kept in mind along with relating the new content to other existing content in the course. Using these considerations, content on using cloud servers seemed most appropriate. This could be related to existing course content on operating systems and Linux and would support networking, web development and capstone courses. In addition, students would gain experience working with the AWS Management Console which would be useful for use of any AWS tools in later courses.

In addition to providing students hands on experience working with cloud infrastructure, content and activities to help students see how the flexibility, scalability, and elasticity of cloud infrastructure can affect strategic IT decision making was also seen as important content to include. Since the course already had content to introduce IT strategy and decision making, this would extend existing course content.

3. THE COURSE

The Computer and Information Technology Department at Miami University offers several degree options. At the bachelor's level, students can earn a degree in Information Technology or a focused degree in Health Information Technology. Several associate degrees are also offered.

Two courses serve to introduce students to the field of Information Technology. These Introduction to IT courses provide students a broad overview of the IT field and are required for all IT degrees. The content of the two courses is structured to allow students to take the courses independently and in either order. The course described in this paper introduces a range of topics including computer architecture, data representations, operating systems, a survey of programming languages, and the wide range of tools used by IT professionals. The course focuses on problem solving in an IT context, including algorithms, analysis, development, testing, and documentation. To highlight the importance of problem solving, an effort is made to use problem solving in as many contexts as possible. For example, the discussions of programming languages and IT tools include the concept of picking the appropriate language and tools as an IT problem.

In addition to introducing students to a number of fundamental technical concepts, an important part of the course is introducing students to the IT profession. Students have many motivations for entering the IT major – previous experience with programming, experience working with computer hardware, recommendations from friends or relatives, interest in writing games, etc. Providing a clear idea of what IT professionals do (and do not do) is also part of the course. This is done through a variety of activities. One activity starts on the first day of class with a discussion of the people, places, and things in IT. These discussions are used to discover student's current knowledge of job titles and work activities (people), work locations and environments (places), and objects (things) that involve IT professionals. These were documented and referred to throughout the semester. The discussion concluded with the instructor identifying any gaps in the items identified by students. A follow up activity asked students to review local IT job postings, select one to post to an online discussion board, and review and comment on items posted by other students. This activity allowed students to start exploring jobs that matched their interest and also exposed them to the variety of skills required for IT positions.

A recurring class activity discussed current events in IT. Each student was assigned a specific class period where they were responsible for finding an article for discussion. Students posted the article in the online Learning Management System (LMS) before class and commented on why they found the article interesting. The student then introduced the article to start an in class discussion. Another student was assigned to post a summary of the in class discussion to the LMS and students were encouraged to continue the discussion online. Selected articles typically cover a broad range of topics, which serves the goal of helping students understand the breadth of the IT profession.

Since this course is a key part of the IT curriculum, it is offered every semester. The particular course session discussed in this paper was taught in a fully face-to-face setting where the class met for two hours and fifty minutes once a week. For the course section discussed in this paper, the initial course enrollment was seventeen (17) students, with fourteen (14) completing the class.

4. CLOUD COMPUTING ACTIVITIES

The course activities that introduced students to cloud concepts occurred towards the end of the course. This allowed the cloud computing material to build on previous material including an introduction to problem solving using the Python programming language and an introduction to the Linux operating system.

The cloud activities were designed to use the same approach as other activities in the class. This approach recognized that introductory classes give students an initial experience with a topic, with many details left for discussion in later courses. To achieve this, assignments were structured with a destination and journey approach. Students were given detailed steps for completing an activity (the destination), but were asked to submit a written discussion of their experience with each step along the journey. For example, in earlier Linux activities, students were given the Linux command needed to complete a task, but needed to consult the documentation (man or info pages) to find needed arguments or interpret output. For some steps, specific writing prompts were provided. For example when they executed the Linux command "cal 9 1752" (which displays the calendar for the month when 10 days were removed as Great Britain switched from the Julian to Gregorian calendar), students were asked if they noticed anything odd, and asked to find an explanation (which could be found in the documentation).

All cloud activities made use of the Amazon Web Services (AWS) cloud so that students could take advantage of the AWS Educate program that provided each student credits that for AWS usage along with training and tutorials. Another value in making use of AWS was that the wide range of companies using AWS included a number that were familiar to the students. These included Netflix, online gaming platforms, and Canvas (the LMS used in the course).

AWS Introduction and Account Setup

The first activity was intended to familiarize students with AWS and complete the account setup process. The activity provided detailed instructions to walk students through the process of applying to AWS Educate and creating an AWS account. After a few days, the students AWS Education applications would be accepted and they received a code they applied to their account to receive a $100 credit. Students did this activity a week before activities that used AWS to allow time to receive the credit and resolve any account issues.

The activity also tasked students with reviewing the products offered by AWS. Students were explicitly directed to the details and pricing of the Elastic Compute Cloud (EC2) since this product would be used in later activities. Students also reviewed the AWS case studies (Amazon EC2, n.d., Case Studies, n.d.) to look for companies they recognized and watched the video from at least on case study.

To make students aware of resources they could use to learn more about AWS, they reviewed the AWS Training and Certification details, and completed the self-paced lab for EC2 (AWS Training, n.d.). To record their participation in this activity, students submitted a one paragraph written discussion covering what they learned about the AWS products, AWS case studies, and AWS training activity.

Using Python on AWS

The next activity using AWS built on an assignment from the Linux operating system module. In the Linux activity, students worked on a Linux server where Python 2 was the default version of Python and had to locate and make use of Python 3 to run a Python program both as a command line argument to the Python interpreter and as a standalone Linux executable.

The Linux assignment introduced the fact that different operating systems use different line terminating characters in text files. Students used Linux tools to identify and fix end of line termination issues with Python files uploaded in a Windows format as part of the process to run the Python program as a standalone Linux file.

The students were asked to complete the same task using AWS, which added the need to create and connect to an EC2 instance using a specified Linux Amazon Machine Instance (AMI). Students had seen the process for creating and connecting to an EC2 instance in the self-paced lab completed in the first activity and were also walked through the process in class, so this process was not completely new to them.

Once the students had started and connected to the EC2 instance, they were told to complete the steps used in their previous assignment. However, the specific Linux AMI that students used was selected to introduce an additional complication to their journey. The specific AMI provided a Linux environment similar to the one used in the previous Linux assignment in having both Python 2 and Python 3 installed, with Python 2 as the default version. However, it did not have the dos2unix tool used to convert text file line encodings installed. When students tried to use this command they got a message that it wasn't installed along with a suggestion on how it could be installed. Students were asked to explore the suggested command and determine what source would be used to install the software. Students then executed the suggested command and completed the activity.

In addition to introducing students to starting and running an AWS EC2 instance, this activity also provided insight into how software updates and installations for an AMI are configured. It extended the Linux content to allow students hands on experience in performing Linux system administrator tasks. Linux system administration was discussed during the Linux module, but due to the potential risk of disruptions to the physical Linux server being used, hands on experience was not possible.

Using AWS to Install a Web Server

The final hands on cloud assignment asked the students to use an EC2 instance to install a web server. This activity served many purposes. First, it showed students that a task they might expect to be complex and time consuming could be greatly simplified by starting from an Amazon AWS EC2 instance. It also introduced students to the security aspects of EC2 instances. Finally, it showed students some of the details behind the web sites they interact with on a daily basis.

For this activity, students started an EC2 instance using a Linux AMI and connected to the instance. They used system administrator access to run two commands to install and start the Apache HTTPD server. They also used process monitoring commands introduced in the earlier Linux module to ensure that the Apache server actually started. For several of these steps, students were given specific reflective prompts that required them to explore on their own. For example, before running the command to install the Apache HTTPD server, they were prompted to determine exactly what would software would be installed, what site would it be installed from, and did they think this was a source they would trust?

Next students used a web browser to access the server. This actually didn't work and resulted in a page not found error. Students were directed to find and review the default security group created for their EC2 instance to see if they could identify the problem. Once they identified that the default security group blocked all HTTP traffic using port 80 on their EC2 instance, they were able to add a security rule allowing inbound HTTP traffic.

When this was successfully completed, students were able to view the default Apache successful installation page. This page provided information for a website administrator about where to add content. Students were directed to create a simple HTML file and load it to their web server. While attempting this, students encountered another issue they needed to understand and address. The issue was a common Linux issue of not having write access to a directory and students needed to use the "sudo" command to overcome this issue.

In addition to providing a screen shot of their HTML file being displayed, students were also asked to reflect on the activity. Specifically, they were asked to explain what the commands provided to install the Apache HTTPD server would do, what source would be used in installing the software, whether they thought it was a trustworthy source, how they resolved the issue with the security group, and how they resolved the final file access problem. Finally, the students were asked to reflect on whether the process was easier or harder than they initially expected.

AWS Cost Estimating Case

The final assignment involving AWS involved a case study that helped students explore how cloud infrastructure like AWS impacts cost estimating and decision making processes for procuring IT infrastructure.

The case involved the need to replace a Linux server that was used to support a data mining course at a college. Students were presented information about the computing and storage capacity of the current server and asked to create a cost estimate for using AWS EC2 to support the class.

This task helped students learn about the different pricing options for AWS products. For the server, EC2 offers on-demand pricing with a set charge per hour of use and also offers reserved instances with different payment options and the potential for significant price reductions if the customer can commit to a year long term (Amazon EC2 Pricing, n.d.). With storage using AWS Elastic Block Store (EBS), there are similar choices based on storage performance (Amazon EBS Pricing, n.d.). Students had to explore the details of these pricing options to prepare the cost estimates.

The initial scenario presented in the case was for a course with an enrollment of 24 students. After students had developed a cost estimate for using AWS to support this scenario, additional scenarios were presented. These included supporting multiple sections of the course in a term and also a single course section with a lower enrollment. Students were also asked to develop their own scenario to explore. Students developed cost estimates for using AWS for all of the scenarios. In addition to a potential cost savings, student were also asked to write a paragraph discussing reasons other than cost for why using AWS might be a better solution.

Exploring the additional scenarios let students see the flexibility and scalability offered by AWS. For example, if a second section of the course was added, the needed AWS resources would be available immediately while it might take several weeks to procure additional physical infrastructure resources.

5. EVALUATION

At the end of the term, a reflective assignment was used to get student feedback on the cloud content and assignments used in the course. The goal of the reflective assignment was to get student input on whether they found the content useful, whether the content should be included in future sections of the course, and suggestions for improving or expanding the content. The first part of the reflective assignment had four survey questions and the second part of the assignment asked student to provide a written reflection on their experience working with AWS.

The survey portion of the assignment asked four questions. The first two used a 5 point Likert scale asking student to agree or disagree with the statements:

- I found the discussion and activities related to Amazon Web Services (AWS) helpful in learning about the cloud computing technology.
- I saw the value of discussing and working with AWS.

The next question asked students to use a 7 point scale to score how much they enjoyed the discussions and assignments related to AWS. The final question asked for a yes/no response about "Should the discussion and assignments using AWS be continued in further sections of this course?"

In the written reflection portion of the assignment, students were asked to address a few specific points and provide any other comments they had. The specific points they were asked to address were:

- What was the most surprising thing you learned about AWS?
- What was your favorite part of learning about AWS?
- How effective were the assignments for learning about AWS?
- What aspects of AWS would you have liked to learn more about?
- If you were teaching the course, what would you do differently for the AWS discussions and assignments?

Of the fourteen (14) active students, eleven (11) submitted the AWS reflection assignment. The three (3) students who did not submit the reflective assignment also did not submit any of the AWS assignments.

The responses to the survey questions indicated that all of the students found the AWS discussions and assignments to be helpful, with the majority (7 of 11) strongly agreeing. Results for the question about whether the student saw the value of the AWS discussions and assignments were similar, with all agreeing and the majority (8 of 11) strongly agreeing.

Responses to the question "How much did you enjoy the discussions and assignments related to AWS (1 = Not Very Much to 7 = Very Much)?" were again all positive, with the average of the responses being 6.1. Finally, all of the students agreed that the AWS content should be continued in future sections of the course.

Students submitted a wide range of comments for the written reflection questions, but several themes could be identified. For the question about what was most "surprising," several students noted that they were surprised to learn that AWS exists and is a significant part of what Amazon does. Most were also surprised by the extensive range of products and options offered through AWS. A number of students were also surprised by the wide range of companies, especially larger companies like Netflix that use AWS.

For the "favorite" part of learning about AWS, almost half of the students mentioned the assignment to setup a web server using AWS. They enjoyed learning more about how a web server works, and were surprised at how easy the process was using AWS. Another theme mentioned by several students was just learning about the wide range of products offered by AWS and the different ways that companies make use of the AWS offerings.

For the question about "How effective" the assignments were, all of the students found them to be effective. Several made positive comments about the breadth of the assignments and the ability to get hands on experience with AWS.

The question about the aspects of AWS they would have liked to learn more about generated a range of responses. Several responses showed interest in more in-depth exploration of the EC2 offering by creating multiple servers that interacted or exploring the capabilities to scale resources up or down. The other responses showed an interest in expanding the breadth of the topic, with specific mention of security and identity management. All of the comments provide good ideas for expanding the cloud content in the course.

For the final question about "If you were teaching the course, what would you do differently for the AWS discussions and assignments?" the most common suggestion was to spend more time on the topic. Two other interesting suggestions were offered. One student suggested that after the introduction to AWS, students could vote on what AWS offerings should be explored in more depth. Another student suggested making more use of the hands on labs provided through the AWS Educate program.

The AWS Educate offering proved useful for the course. A couple of students had minor issues with the registration process, but these were resolved and did not prevent students from completing the AWS Introduction assignment on time. A couple of students reported problems in completing the self-paced lab on EC2, but these issues were easily resolved. One of the perceived values of the AWS Educate program is the $100 credit provided to students. At the end of the course, students were reminded of the credit and the AWS Educate training resources and encouraged to explore more AWS products.

6. CONCLUSION AND NEXT STEPS

The main conclusions that can be drawn from the assessment of the cloud content and assignments piloted in this course are that students saw the value in learning about cloud computing and found the specific discussions and assignments helpful in learning about cloud computing. The content clearly helped students understand more of the "magic" behind software and services they use on a daily basis.

The written comments students offered expressing surprise on learning about AWS, its scale, and their recognition of a number of the companies using it were also expressed in class discussions. This supports that idea that IT students need to be aware of the impact that cloud computing is having, and that size of AWS makes it a good choice for introductory activities.

From the instructor's perspective, the student feedback from the reflective assignment along with their enthusiasm shown during in class discussions and in submissions for the AWS related assignments validates the decision to add this content to the course. The lack of problems, breadth of offerings, and supporting labs and tutorials confirmed that AWS Educate was a good platform to use for these activities.

For future offerings of the course, the cloud content will be extended. The most likely addition would be content that demonstrates the elasticity of cloud offerings. Letting students select additional content to explore is also being considered, possibly as a group assignment with each group exploring a different topic and sharing what they learn with the rest of the class.

Another idea that is being considered is to combine the Linux and cloud computing content. There is a natural affinity between the two, and the wide range of EC2 instance types and Linux and AMIs would allow assignments to explore a wider variety of Linux flavors and server configurations than is possible with the physical server currently used in the Linux module. Additionally, the availability of Windows AMIs would allow students to see a wider range of operating systems (Windows Server on AWS, n.d.).

A couple of other long term ideas are also being developed. One is to identify other courses in our curriculum that could make use of AWS Educate resources and cover topics in the cloud content of the introductory course that will prepare students to make further use of AWS offerings in the later course. The idea of expanding the content into a full course is also worth exploring. Finding room in the curriculum may be a challenge, but covering the cloud in a special topics course could be an interim solution.

In conclusion, while adding new content to a course is a risk, with unpredictable results, the outcome in this case exceeded the instructor's expectations.

7. REFERENCES

Amazon EBS Pricing. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/ebs/pricing/.

Amazon EC2 Pricing. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/ec2/pricing/.

Amazon EC2 – Virtual Server Hosting. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/ec2/.

AWS Educate. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/education/awseducate/.

AWS Training. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/training/intro_series/.

Case Studies & Customer Success Stories, Powered by the AWS Cloud. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/solutions/case-studies/.

Hwang, D., Pike, R., & Mason, D. (2016). The Development of an Educational Cloud for IS Curriculum through a Student-Run Data Center. *Information Systems Education Journal*, 14(1), 62-70.

Izrailevsky, Y. (2016). Completing the Netflix Cloud Migration. Retrieved June 14, 2017, from https://media.netflix.com/en/company-blog/completing-the-netflix-cloud-migration.

Kelleher, K. (2015). Why This Is the Most Important New Technology in Decades. *Time.* Retrieved June 14, 2017, from http://time.com/3856963/2015-the-cloud/.

Krazit, T. (2017). AWS Revenue up 42 percent to $3.66 billion in Q1 2017, operating income reaches $890 million. Retrieved June 14, 2017, from https://www.geekwire.com/2017/aws-revenue-42-percent-3-66-billion-q1-2017-operating-income-reaches-890-million/.

Lawler, J. (2011). Cloud Computing in the Curricula of Schools of Computer Science and Information Systems. *Information Systems Education Journal*, 9(2), 34-54.

Mew, L. (2016). Information Systems Education: The Case for the Academic Cloud. *Information Systems Education Journal*, 14(5), 71-79.

Mills, T. (2011). Tradition meets technology: top universities using Apps for Education. Retrieved June 14, 2017, from https://googleblog.blogspot.com/2011/09/tradition-meets-technology-top.html.

Morgan, T. (2014). A Rare Peek Into The Massive Scale of AWS. Retrieved June 14, 2017, from http://www.enterprisetech.com/2014/11/14/rare-peek-massive-scale-aws/.

Pike, R, Pittman, J., & Hwang, D. (2017). Cloud-based Versus Local-based Web Development: An Experimental Study in Learning Experience. *Information Systems Education Journal*, 15(4), 52-68.

Rehman, M., Boles, J., Hammond, M., & Sakr, M. (n.d.). A Cloud Computing Course: From Systems to Services. In *Proceedings of the 46th ACT Technical Symposium on Computer Science Education* (pp. 338-348). New York, NY: ACM. doi:10.1145/2676723.2677298

Serapiglia, A. (2013). LINUX, Virtualization, and the Cloud: a hands-on student introductory lab. *Information Systems Education Journal*, 11(5), 57-64.

Windows Server on AWS. (n.d.). Retrieved June 14, 2017, from https://aws.amazon.com/windows/.

Grit and the Information Systems Student: A Discipline-Specific Examination of Perseverance and Passion for Long Term Goals

Nita G. Brooks
nita.brooks@mtsu.edu

Scott J. Seipel
scott.seipel@mtsu.edu

Department of Computer Information Systems
Middle Tennessee State University
Murfreesboro, TN USA

Abstract

Grit has been highlighted in recent research as a distinct trait believed to be associated with performance and success factors above and beyond those explained by cognitive ability. It focuses on the dedication required to meet long-term goals and is represented by two subscales: consistency of interest and perseverance of effort. The overall goal of the current study is to understand the operation of the grit construct and its relationship with key demographic factors for information systems students specifically. Data was collected from 176 information systems undergraduate and graduate students at a public university in the southeastern United States. Analysis was conducted using structural equation modeling. Individual models were created and examined that included grit and key factors shown in previous research as related to grit: age, GPA, and gender. Additional factors were included related to employment status (full-time, part-time, unemployed) and academic classification (freshman, sophomore, junior, senior, and graduate student). Findings from the analysis of the grit structure in conjunction with these different factors indicate that grit and employment status are related. Individuals that specified they were employed full-time had higher levels of grit. For this group of students, findings revealed some inconsistencies with previous research and the relationship of grit to the additional factors studied, highlighting the need for discipline-specific examinations of construct. A detailed discussion of the results is provided along with implications and suggestions for future research.

Keywords: grit; information systems students; Grit-S; long-term goals; perseverance

1. INTRODUCTION

At the center of research related to the information systems (IS) discipline and higher education, there is the goal of understanding the student to the greatest level possible in hopes of being able to foster progression through academic programs as well as to improve the likelihood of future career success. Research has indicated "30% of students who entered college in the fall of 2014 did not return in the second year" (Lee & Stewart, 2016 p. 2). Additionally, information systems remains a field where supply is not meeting demand in relation to providing individuals to sustain and support the workforce (White, 2016). Academic programs struggle with recruiting and retaining students (Hunsinger, Land, & Chen, 2012) in conjunction with these other factors. As such, it is necessary to continue and strive to better know our current students. One avenue that has not been explored for the IS discipline is to understand information systems students' level of grit.

Over the past decade, research related to the concept of grit has highlighted usefulness of the construct as a predictor of performance and success in different areas (career, academics, etc.). Grit is a trait-like factor defined as "perseverance and passion for long-term goals" (Duckworth, Peterson, Matthews, & Kelly, 2007 p. 1087). It is operationalized as two factors representing consistency of interest and perseverance of effort. Von Culin, Tsukayama, and Duckworth (2014) define consistency of interest as "abiding focused interests over time" (p. 308) and perseverance of effort as a "tendency toward a sustained effort" (p.308). A driver in this area of research has been to find a response to the question – "why do some individuals accomplish more than others" (Duckworth, et al. 2007, p. 1087). Previous studies have examined the role of intelligence and the Big Five personality traits in attempting to understand and answer this question (Credé, Tynan, & Harms, 2016). Grit has been shown to extend the explanatory capabilities of success and performance models beyond these traditional factors. It has also been shown as related specifically to retention (Duckworth & Quinn, 2009).

To begin our examination, the following section provides an overview of the grit literature. Special attention is given to the items that are examined in this study to provide a foundation on which to examine grit in information systems students.

2. LITERATURE REVIEW

Previous research related to understanding performance and success has often looked to personality traits as ways of understanding why individuals differ related to these types of outcomes (Siebert & Kraimer, 2001). Determining reasons why some people are more successful than others has driven a majority of educational and workforce research efforts.

From an education perspective, success is often measured by the progression of an individual through the required stages of an academic program; the final result of which is the completion of a degree. The examination of personality traits in relation to academic success has also shown a connection. In fact, "the contribution of personality traits to academic achievement may be as great as or greater than that of intelligence" (Willingham, 2016 p.30).

In an effort to expand what is understood regarding factors that impact individual success and performance, grit was operationalized as a unique construct representing an individual's perseverance and passion for long-term goals (Duckworth, et al. 2007). It is a concept that is closely, but distinctly, related to the trait of conscientiousness. While conscientiousness focuses on "short-term intensity" (Duckworth, et al. 2007 p. 1089), grit emphasizes the long-term by examining two subscales: consistency of interest and perseverance of effort. Research specifically related to conscientiousness has shown significant relationships with the trait and measures of success, including factors related to an individual's career (pay, promotion, satisfaction) (Thomas, Eby, Sorensen, & Feldman, 2005).

Since its introduction, grit has garnered much research attention. It has been related to GPA, success in national competitions, such as the National Spelling Bee, as well as other areas that require deliberate practice over a period of time (Credé, Tynan, & Harms, 2016; Willingham, 2016). Specifically, it has been shown that students with higher levels of grit outperform their peers with lower levels of grit evidenced by higher GPAs (Duckworth, et al. 2007).

Age has also been linked to grit in previous studies. In a study examining grit scores of over 1500 individuals (25 and over), it was found that older adults have higher levels of grit (Duckworth & Quinn, 2009). While the study did not, intentionally, include younger individuals, the impact of age was significant in this case, but when looking across the literature results are mixed (Credé, et al. 2016).

Gender has also been examined in numerous studies focused on understanding grit and other trait-like factors. Interestingly, across various samples, gender's relationship to grit has been weak and somewhat mixed (Credé, et al. 2016). In relation to other personality traits, men and women often differ significantly (Maestripieri, 2012). In examining the related construct of conscientiousness, for example, women often score higher than men on some aspects, but, again, the findings are mixed (Weisberg, DeYoung, & Hirsh, 2011).

An additional factor we are considering in this study relates to academic classification and where students are in their program of study. While this has not been specifically examined in relation to grit, previous research has found that "more educated adults were higher in grit than were less educated adults of equal age (Duckworth, et al., 2007 p. 1091).

The following section details the methodological approach taken in examining grit and the concepts just described for information systems students. General demographic information is summarized and detailed model analyses presented.

3. METHODOLOGY

To assess grit, the eight-item Short Grit Scale (Grit-S) was utilized (Duckworth & Quinn, 2009). In this short scale version, 4 items each are used to determine the individual's consistency of interest and perseverance of effort (Table 1). All items were measured on a 7-point Likert-type scale with 1 representing "Very untrue of me" and 7 representing "Very true of me". Consistency of interest was measured on a reversed scale, where higher values indicate more inconsistency. These scores were inverted prior to analysis for improved interpretability.

Additional information collected in the survey included demographic data on gender, age, employment, and year (academic classification) in school. Respondents were also requested to self-report their current overall GPA as well as their GPA in IS courses only. This data was collected in nine ordinal categories rather than as a raw value (Table 2).

Data was collected from students at a large public university in the southeastern United States. Surveys were distributed in graduate and undergraduate courses offered by the Department of Computer Information Systems housed in the university's college of business. The courses targeted were those required as part of the information systems degree programs.

A total of 196 students took part in the voluntary survey; this represents 54.3% of the students enrolled in the program at the time of the data collection. Although there was a possibility of students seeing the survey in multiple courses, they were requested to only take part once. From the 196 responses, nine were removed from the sample due to lack of significant completion or invariable responses (i.e. one response repeated throughout the questionnaire). Of the remaining 187 responses, 11 were subsequently removed due to incomplete responses to the eight grit items resulting in a sample consisting of 176 students.

A total of eight courses were surveyed in the study, with the level of the course noted as a proxy for how far a student has progressed in their academic program. Two courses were surveyed at the sophomore (60 responses) and junior level (32 responses), three at the senior level (67 responses), and one at the graduate level (17 responses). Descriptive statistics on demographic information, including academic classification, are shown in Table 2. From the confluence of course and academic position information, it was determined that approximately 57% of the respondents were in courses at a level consistent with their classification (e.g. a student in a junior level course was a junior by classification). Due to the utilization of information systems as a secondary "career saving" major, this was considered a potentially important distinction in determining how program progress relates to grit.

4. ANALYSIS

An initial confirmatory factor analysis was performed to determine the degree of fit of the pure grit construct to the population of information systems students. Determination of fit is based on the standard measures of the χ^2 statistic, the root mean square error of approximation (RMSEA; Steiger & Lind, 1980), the comparative fit index (CFI; Bentler, 1990), and the non-normed fit index (NNFI; Tucker & Lewis, 1973). According to Hu and Bentler (1999), a value of .06 or below is considered an acceptable fit based on the RMSEA, while the comparative values for the CFI and NNFI are .90 or more (with .95 or greater preferred). Analyses were performed utilizing R (R Core Team, 2013) and the lavaan package (Rosseel, 2012).

Based on a positive overall fit, the next step is to determine if any of the factors captured in the study are associated with the grit level of the respondents. Factors under consideration include the standard demographic variables like gender and age, along with employment status. We also extend the research to academic classification and the course in which the survey was performed. The analysis of these factors is performed via a series of measurement models with increasing restrictions on the components of the model that are allowed to vary in each group. Model 1, the baseline model, incorporates the groups into the model with no restriction other than equivalent factorial structure. A good fit of Model 1 would indicate configural invariance among the groups. Model 2, which includes the factor structure constraint from Model 1, adds the restriction that factor loadings are equivalent among the groups. The fit of Model 2 would indicate metric invariance and allows for the investigation of group differences in grit or its subscales. Model 3 adds a requirement for equal intercepts to the requirements of Model 2 and is an indication of scalar invariance. Model 4 requires the equivalence of error variances among the groups in addition to prior restrictions and is often referred to as strict invariance; it is not necessary to achieve proper fit in Model 4 in order to compare scores. In this analysis, Model 5 represents the equivalence of factor variance/covariance structures among the groups, and is incremental to the requirements of Model 4. Model 6, which is the final model in this analysis, considers whether factor means can be considered equal among the groups. It should be noted that this model will be tested as a marginal change from Model 4; Model 5 fit is not required.

As the results of each model are incremental, a determination can be made concerning the extent to which the grit construct differs among these groups (i.e. when the marginal change produces an ill-fitting model, then the prior model provides the extent to which the groups do not vary). In evaluating these models, the Akaike Information Criterion (AIC; Akaike, 1974) and McDonald's non-centrality index (NCI; McDonald, 1989) will also be used due to their applicability to nested models. For nested models, lower values of AIC generally indicate a better fit. Cheung and Rensvold (2002) recommend marginal changes in excess of -.01 and -.02 for the CFI and NCI measures respectively to move to a more restricted model. A good explanation of the process of testing measurement invariance can be found in Milfont and Fischer (2010).

Following the results of measurement invariance analysis, the relationship of the grit factor model to quantitative variables age, overall GPA, and major GPA are evaluated. Subsequently, composite scores are determined for each component of grit for each respondent. Those scores are subjected to the analyses of variance/covariance to determine the effect of variables that had a significant relationship to the individual subscales of grit.

Results

The internal consistency of the overall Grit-S scale as measured by Cronbach's Alpha was .71. Measures for the individual subscales of consistency of interest and perseverance of effort were .61 and .63 respectively. A maximum-likelihood confirmatory factor analysis was run on the sample on the first-order latent variable of consistency of interest and perseverance of effort loading on the second order latent grit factor. Indications of overall model fit were strong, with χ^2 = 19.62 (19 *df*, *p* = .418), RMSEA = .014 (90% CI = .000-.068), CFI = .997, and NNFI = .995. All *p*-values of estimated parameters were less than .001. The observed correlation between subscales was .448. The ratio of observations per estimated parameter was greater than 9 to 1, sufficient based on the criterion of exceeding 5 to 1 from Bentler and Chou (1987).

The invariance of the grit factor model was evaluated as it related to the gender of the respondent. Models 1 through 6 were fitted with corresponding fit statistics shown in Table 3. Based on the results from Model 1, it can be concluded that the overall fit of the grit model to information systems students was acceptable when gender is brought in as a mitigating factor. The results show that increasing restrictions are not significantly detrimental to the model's fit. All models show acceptable fit levels and marginal changes to AIC, CFI, and NCI are within acceptable values at all increments. Given these results, it can be concluded that gender plays no role in the measurement of grit in this population.

The structure of the grit factor model relative to the self-identified employment groups on the survey was evaluated next. The baseline model with the inclusion of the employment factor indicated strong fit (Table 4). Subsequent nested models indicated a noticeable degradation of fit from Model 3 (scalar invariance) to Model 4 (strict invariance), signifying a difference in residual error in the groups. These findings show that there is sufficient basis to evaluate mean composite grit scores among employment groups in a follow-up analysis.

The evaluation of the grit factor model to the position in program (i.e. level of course in which the measurement was made) and academic classification (i.e. Sophomore, Junior, Senior) was not possible due to non-convergence of confirmatory factor analyses when these groups where separately analyzed. This is likely due to relatively small sample sizes in certain groups (e.g. 17 observations in the 6000 level course and 15 observations at the sophomore level). Convergence was achieved by imposing parameter restrictions on the model, but this was considered to be too assumptive to allow for conclusive analysis.

To evaluate the effect of age on the measurement of grit, the model used in the initial confirmatory factor analysis was modified to include age as a covariate to both subscales. Model fit was acceptable but had degraded from the initial model: χ^2 = 35.613 (25 *df*, *p* = .078); RMSEA = .050 (90% CI = .000-.084); CFI = .949; NNFI = .926. The *p*-values of all estimated parameters were at or less than .002 except for the parameter estimates for age for both subscales. The *p*-value for the age coefficient for consistency of interest was .052 and for perseverance of effort was .804. As the effect of age may have been carried through the correlation of the two subscales, it was decided to drop the least significant relationship of age to perseverance of effort. The subsequent model showed marginally better fit: $\chi^2 = 35.671$ (26 *df*, $p = .098$); RMSEA = .047 (90% CI = .000-.081); CFI = .953; NNFI = .935. Importantly, all parameter estimates had *p*-values at or less than .002 including the parameter for the relationship of age to consistency of interest.

Numerical estimates were determined for the overall and within major GPAs utilizing the midpoint of each GPA category in the survey. These variables were included in the initial grit model under both of the grit subscales. The fit of this All GPA model was marginally within the acceptable range (Table 5). All parameters relating each GPA to each subscale were insignificant (*p*'s > .138). Removing insignificant parameters sequentially led to a Reduced Model where both GPAs were only related to consistency of interest ($p < .001$). However, the fit of the overall grit model had degraded significantly such that the model no longer fit acceptably. Interestingly, the problem appeared to be related to redundant information being passed along these GPA variables. Individuals with high overall GPA were also probably the individuals with high within-major GPA. From this analysis, the best model included overall GPA to consistency of interest (Table 5).

As a follow-up to the results of the measurement invariance analysis and covariates, composite scores for the individual subscales of grit were calculated to provide additional insight into their relationship with employment status and age. Separate ANOVAs were performed on each of the subscales to determine the significance and direction of employment effects.

At least one difference was significant among the mean consistency of interest composite scores for the different levels of employment status ($p = .0030$). Using Tukey's honest significant difference test (HSD) for pairwise comparisons, it was determined that students who are employed full-time had a mean composite score .349 higher (95% CI: .051-.646) than part-time employed students, and a mean composite score .458 (95% CI: .124-.791) higher than students who were not employed. The difference between part-time and non-employed students was not significant.

For the perseverance of effort subscale, there was at least one significant difference in the mean composite score among the employment levels ($p = .0073$). As with the other subscale, Tukey's HSD found differences between full-time and part-time employed students (d = .277; 95% CI: .010-.545) and full-time and non-employed students (d = .384; 95% CI: .084-.684); the mean scores of part-time and non-employed students were not significantly different. Assumptions of the ANOVA were again reasonable based on an analysis of the residuals.

Findings indicated that age was related to the measure of grit. There is also reason to believe that age and employment might be correlated. An ANCOVA was performed to analyze mean composite grit scores for differences among levels of employment status after taking into account the age of the respondent. For consistency of interest, both employment status ($p = .0275$) and age ($p = .0165$) were significantly related to the mean composite score. Interestingly, the inclusion of age did not sufficiently reduce the unexplained error of the model enough to make any major changes to the results from Tukey's HSD. The 95% confidence intervals for the differences between full-time/part-time and full-time/non-employed were very close to the results without age taken into account. On the perseverance of effort subscale, the results of the ANCOVA showed age did not have a significant effect ($p = .062$) on the mean composite score. Assumptions appeared to be reasonable under both tests.

5. DISCUSSION

In this examination of grit, the focus was specifically on information systems students and the relationship of grit with key demographic variables. Findings revealed some consistencies and differences when compared to previous research on the trait.

In alignment with existing findings, gender did not impact the grit model (Duckworth & Quinn, 2009). For this group, males and females see grit the same. While the sample consisted of a majority of male students (77%), it was generally representative of the profession where females account for approximately 25% of all computer-related occupations (Ashcraft, McLain, & Eger, 2016). The sample was also representative of the student body comprising the information systems program (Table 2). The findings imply that grit does not vary across gender and that men and women have the same level of grit. With the long-term focus on grit, it seems that men and women are not distinct in their levels of enduring interests and the level of effort they sustain.

In examining grit and adding age to the model, we found age was related more to consistency of interest than perseverance of effort. Age has been found to be significantly related to grit in previous studies (Duckworth and Quinn, 2009) but has been shown in other cases to have only a slight correlation with the overall grit measure (Credé, et al., 2016). As findings across studies reveal somewhat different relationships, further examination is warranted. The examination of age as related to the sub-scales of grit may provide more information on the importance of considering age of the individual in understanding this trait. This could involve looking at other academic disciplines and providing the opportunity to compare across groups; it could also involve extending the examination to include individuals that work in the IS profession.

When examining grit with the addition of the GPA items in the model, findings revealed that GPA was more related to consistency of interest for this group of information system students, while the overall grit measure was found to be related to GPA in a previous study (Duckworth, et al., 2007). More recently, perseverance of effort was found to be a "superior predictor of GPA" (Duckworth & Quinn, 2009), which is counter to our findings. It appears that, in this case, information systems students do differ from other populations.

The findings also indicate that information systems as an academic discipline is tightly aligned with practice. For this group of students, grit was associated with employment, with students employed full-time having the highest levels of grit when compared to their counterparts. Previous research has noted that some of the matters often attributed with individuals not completing academic programs relates to financial and work-life balance issues (Lee & Stewart, 2016). This finding indicates that there may be an associated level of grit and the ability to successfully manage multiple responsibilities.

As a final point of discussion, it is necessary to consider the implications of the findings and what it means for those in higher education and practice. Part of the discussion of grit, as well as other personality-related traits, surrounds the ability to teach or alter grit. For example, if a certain level of grit is associated with success in a particular area, whether it is in an academic program or in a job or career, is it possible to alter "grittiness"? (Willingham, 2016). Answering this question is beyond the scope of the current study, but it does warrant consideration when examining the construct and the purpose of understanding it.

6. LIMITATIONS AND FUTURE RESEARCH

It is necessary to address the limitations of the current study and paths for improving and expanding research in this area. One limitation involves the use of a cross-sectional study design. It would be beneficial to collect data at multiple points in time during an individual's progression through an academic program. An example could be to collect data upon entry into the university and program followed by each milestone achieved as the student moves from being a freshman to sophomore, etc. The collection of data longitudinally would assist in understanding the role of grit's impact on success and whether there are shifts in individual levels of grit.

Additionally, the data was collected from students at one university, which could limit generalizability of the findings. Outcomes have been shown to differ across institution types. For example, the National Center for Education Statistics indicates that the graduation rate for public universities is 58% compared to private non-profit schools at 65% over the same six-year time period (Lee & Stewart, 2016). Expanding the research to include individuals at other institutions as well as different types (public and private universities) would provide additional support and detail in understanding grit in students in information systems programs.

Another issue presented in this analysis involved the self-reporting of GPA. Since no identifying information was collected, it was not possible to connect official GPA information to the individual participants. In an effort to better understand the implications for the current study, the researchers obtained general information on all information systems majors and compared the results to what was reported by the participants. As provided in Table 2, there were differences. Future research should aim at collecting official GPAs to further understand the results from this initial analysis.

The findings of this study indicated that individuals with full-time jobs had higher levels of grit when compared to individuals in part-time positions or those that stated they were unemployed. Extending research around factors related to work situations and other work-life factors could provide additional insight and help answer questions such as, do gritty full-time employees go after a degree to improve themselves or do gritty students try to take on responsibilities beyond education? Interesting factors to consider would include number of children, marital status, income level, and whether or not the individual is the first in their family to attend college. It could also prove important to know whether the individuals that were employed were in an information systems role or a job unrelated to the field. If individuals are employed in information systems, it would stand to reason that their career decisions are driven not only by extrinsic items such as pay but by their intrinsic motivations to be a part of the IS profession.

As a final recommendation for future research, it seems necessary to examine the role that grit plays in determining key outcomes for not only information systems students but for students in other disciplines. The findings of this initial study provided interesting results that did not align with previous research examining grit. Determining whether grit differs for students in different majors could provide insight into the types of students that are drawn to certain areas and whether or not the students are likely to succeed. Knowing more about traits associated with the individuals that comprise different disciplines could also assist in advising, increasing enrollment, and defining program components that adequately and appropriately align with student characteristics.

7. CONCLUSION

In summary, this study examined grit for information systems students and investigated the impact of certain demographic variables on overall grit and the subscales of perseverance of effort and consistency of interest. After running multiple analyses, findings revealed that employment status and grit are related, which had not previously been examined. Additionally, the data collected from information systems students exposed relationships that were not completely consistent with existing research. These findings warrant the continued examination of the concept of grit giving consideration to the discipline. Results also highlight the need for further examination of employment-related factors for information systems students.

8. REFERENCES

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716–723.

Akos, P. & Kretchmar, J. (2017). Investigating grit as a non-cognitive predictor of college success. *The Review of Higher Education*, 40(2), 163-186.

Ashcraft, C., McLain, B., Eger, E. (2016). Women in tech: the facts., *The National Center for Women in IT*. Retrieved from https://www.ncwit.org/sites/default/files/resources/ncwit_women-in-it_2016-full-report_final-web06012016.pdf.

Bentler, P.M. (1990). Comparative Fit Indexes in Structural Models. *Psychological Bulletin*, 107, 238-246.

Bentler, P.M. & Chou, C.P. (1987). Practical issues in structural modeling. *Sociological Methods & Research*, 16, 78-117.

Carneval, A.P., Smith, N., Melton, M., & Price, E.W. (2015). Learning while earning: The new normal. Center on Education and the Workforce, McCourt School of Public Policy, Georgetown University.

Cheung, G.W. & Rensvold, R.B. (2002). Evaluating Goodness-of-Fit Indexes for Testing Measurement Invariance, *Structural Equation Modeling: A Multidisciplinary Journal*, 9(2), 233-255.

Credé, M., Tynan, M.C., & Harms, P.D. (2016). Much ado about grit: A meta-analytic synthesis of the grit literature. *Journal of Personality and Social Psychology*, June, 1-20.

Duckworth, A.L., Peterson, C., Matthews, M.D., Kelly, D.R. (2007). Grit: Perseverance and passion for long-term goals. *Journal of Personality and Social Psychology*, 92(6), 1087-1101.

Duckworth, A.L. & Quinn, P.D. (2009). Development and Validation of the Short Grit Scale (Grit-S). *Journal of Personality Assessment*, 91(2), 166-174.

Hu, L. & Bentler, P.M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6(1), 1-55.

Hunsinger, D.S., Land, J., & Chen, C.C. (2012). Enhancing students' loyalty to the information systems major, *International Journal of Information and Communication Technology Education*, 6(1), 81-95.

Lee, P. & Stewart, D. (2016). Women in IT jobs: it is about education, but also about more than just education. Retrieved from https://www2.deloitte.com/global/en/pages/technology-media-and-telecommunications/articles/tmt-pred16-tech-women-in-it-jobs.html.

Maestripieri, D. (2012, January). Gender differences in personality are larger than previously thought. *Psychology Today*. Retrieved from https://www.psychologytoday.com/blog/games-primates-play/201201/gender-differences-in-personality-are-larger-previously-thought.

McDonald, R.P. (1989). An index of goodness-of-fit based on non-centrality. *Journal of Classification*, 6, 97-103.

Milfont, T.L. & Fischer, R. (2010). Testing measurement invariance across groups: Applications in cross-cultural research. *International Journal of Psychological Research*, 3(1), 111-121.

Ng, T.W.H., Eby, L.T., Sorensen, K.L, & Feldman, D.C. (2005). Predictors of objective and subjective career success: a meta-analysis. *Personnel Psychology*, 58, 367-408.

R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from http://www.R-project.org/.

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, 48(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

Seibert, S.E. & Kraimer, M.L. (2001). The five-factor model of personality and career success. *Journal of Vocational Behavior*, 58, 1-21.

Steiger, J.H. & Lind, J.C. (1980, May). *Statistically-based tests for the number of common factors*. Paper presented at the Annual Spring Meeting of the Psychometric Society, Iowa City, IA.

Tucker, L.R. & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. *Psychometrika*, 38, 1-10.

Von Culin, K.R., Tsukayama, E., & Duckworth, A.L (2014). Unpacking grit: motivational correlates of perseverance and passion for long-term goals. *The Journal of Positive Psychology*, 9(4), p. 306-312.

Weisberg, Y.J., DeYoung, C.G., & Hirsh, J.B. (2011). Gender differences in personality across the ten aspects of the big five. *Frontiers in Psychology*, 2 (178), 1-11.

White, S.K. (2016, March). The future looks bright for IT workers. *CIO Magazine*. Retrieved from http://www.cio.com/article/3046177/careers-staffing/the-future-looks-bright-for-it-workers.html.

Willingham, D.T. (2016). Ask the cognitive scientist: "Grit" is trendy, but can it be taught?. *American Educator*, 40(2), 28-32.

Appendix

Table 1: Grit Items

Item	Subscale	Statement
1	CI	I often set a goal but later choose to pursue a different one.
2	CI	New ideas and projects sometimes distract me from previous ones.
3	CI	I have been obsessed with a certain idea or project for a short time but later lost interest.
4	CI	I have difficulty maintaining my focus on projects that take more than a few months to complete.
5	PE	I finish whatever I begin.
6	PE	Setbacks do not discourage me.
7	PE	I am a hard worker.
8	PE	I am diligent.

Note: CI=Consistency of Interest; PE=Perseverance of Effort

Table 2: Demographics

Variable	Sample	Sample Percent	Percent in Program
Year			
Freshman	0	0.0%	0.3%
Sophomore	15	8.6%	8.3%
Junior	42	24.0%	18.0%
Senior	82	46.9%	46.3%
Graduate	36	20.6%	27.1%
Gender			
Male	137	77.8%	78.7%
Female	39	22.2%	21.3%
GPA overall			
Below 2.00	0	0.0%	0.8%
2.00-2.24	2	1.1%	3.9%
2.25-2.49	6	3.4%	6.1%
2.50-2.74	18	10.3%	10.0%
2.75-2.99	18	10.3%	10.2%
3.00-3.24	32	18.4%	16.3%
3.25-3.49	29	16.7%	17.2%
3.50-3.74	27	15.5%	15.2%
3.75-4.00	42	24.1%	20.2%
GPA within major			
Below 2.00	1	0.6%	3.9%
2.00-2.24	0	0.0%	1.1%
2.25-2.49	5	2.9%	3.6%
2.50-2.74	9	5.2%	5.8%
2.75-2.99	5	2.9%	5.0%
3.00-3.24	33	19.0%	10.8%

3.25-3.49	28	16.1%	14.7%
3.50-3.74	22	12.6%	19.4%
3.75-4.00	71	40.8%	35.7%
Employment			
Full-time	52	29.5%	
Part-time	77	43.8%	
Age			
19-21	42	24.4%	
22-24	48	27.9%	
25-27	27	15.7%	
28-30	25	14.5%	
31-33	8	4.7%	
34-36	8	4.7%	
37-39	6	3.5%	
40-42	2	1.2%	
43-45	0	0.0%	
46-48	3	1.7%	
49 or more	3	1.7%	

Table 3: Goodness of Fit Statistics for Model Testing

Measurement Invariance across Gender

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	48.46	38	0.119	0.056	4860.4	0.949	0.971	0.925
2	55.88	44	0.108	0.055	4855.9	0.942	0.967	0.927
3	63.00	50	0.103	0.054	4851.0	0.937	0.964	0.929
4	72.10	58	0.101	0.053	4844.1	0.932	0.961	0.934
5	73.57	61	0.130	0.048	4839.5	0.939	0.965	0.944
6	73.91	60	0.107	0.051	4841.9	0.933	0.961	0.937

Comparison of Nested Models

Models	$\Delta\chi^2$	Δ*df*	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	7.42	6	0.284	-0.001	-4.6	-0.007	-0.004	0.001
2 to 3	7.12	6	0.310	-0.001	-4.9	-0.006	-0.003	0.003
3 to 4	9.10	8	0.334	-0.002	-6.9	-0.005	-0.003	0.005
4 to 5	1.47	3	0.689	-0.004	-4.5	0.007	0.004	0.010
4 to 6	1.81	2	0.405	-0.001	-2.2	0.001	-0.004	0.003

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant covariance matrices; Model 6 = Model 4 plus invariance means.

Table 4: Goodness of Fit Statistics for Model Testing

Measurement Invariance across Employment Status

Model	χ^2	*df*	p-value	RMSEA	AIC	CFI	NCI	NNFI
1	52.15	57	0.657	0.000	4849.7	1.000	1.000	1.036
2	64.72	69	0.624	0.000	4838.3	1.000	1.000	1.027
3	77.82	81	0.580	0.000	4827.4	1.000	1.000	1.017
4	113.50	97	0.121	0.054	4831.0	0.916	0.954	0.927
5	127.69	103	0.050	0.064	4833.2	0.874	0.932	0.897
6	127.88	101	0.037	0.067	4837.4	0.863	0.926	0.886

Comparison of Nested Models

Models	$\Delta\chi^2$	Δ*df*	p-value	Δ RMSEA	Δ AIC	Δ CFI	Δ NCI	Δ NNFI
1 to 2	12.57	12	0.401	0.000	-11.4	0.000	0.000	-0.010
2 to 3	13.10	12	0.362	0.000	-10.9	0.000	0.000	-0.010
3 to 4	35.68	16	0.003	0.054	3.7	-0.084	-0.046	-0.090
4 to 5	14.19	6	0.028	0.010	2.2	-0.042	-0.022	-0.030
4 to 6	14.38	4	0.006	0.014	6.4	-0.053	-0.006	-0.041

Note: RMSEA = root mean square error of approximation; AIC = Akaike Information Criterion; CFI = comparative fit index; NCI = McDonald's non-centrality index; NNFI = non-normed fit index. Model 1 = equality of overall structure; Model 2 = Model 1 plus invariant loadings; Model 3 = Model 2 plus equivalent intercepts; Model 4 = Model 3 plus invariant residuals; Model 5 = Model 4 plus invariant covariance matrices; Model 6 = Model 4 plus invariance means.

Table 5: Goodness of Fit Statistics for Models Testing

Overall and Within Major GPA in Grit Models

Model	χ^2	*df*	p-value	RMSEA	CFI	NNFI
All GPA	52.85	32	0.012	0.061	0.935	0.908
Reduced Model	136.27	34	0.000	0.131	0.679	0.575
Overall GPA with CI	28.58	26	0.318	0.025	0.986	0.981
Major GPA with CI	21.58	26	0.712	0.000	1.000	1.031

Note: RMSEA = root mean square error of approximation; CFI = comparative fit index; NNFI = non-normed fit index. CI = Consistency of Interest. All GPA model = overall GPA and major GPA related to both subscales; Reduced Model = overall GPA and major GPA related to Consistency of Interest; Overall GPA with CI = overall GPA related to Consistency of Interest; Major GPA with CI = major GPA related to Consistency of Interest.

A Comparison of Key Concepts in Data Analytics and Data Science

Kirby McMaster
kmcmaster@weber.edu

Brian Rague
brague@weber.edu

Computer Science
Weber State University
Ogden, UT 84408

Stuart L. Wolthuis
stuartlw@byuh.edu
Computer & Information Sciences
Brigham Young University-Hawaii
Laie, HI 96762

Samuel Sambasivam
ssambasivam@apu.edu
Computer Science
Azusa Pacific University
Azusa, CA 91702

Abstract

This research study provides an examination of the relatively new fields of Data Analytics and Data Science. We compare word rates in Data Analytics and Data Science documents to determine which concepts are mentioned most often. The most frequent concept in both fields is *data*. The word rate for *data* is more than twice the next highest word rate, which is for *model*. This contrasts sharply with how often the word *data* appears in most Mathematics books. Overall, we observed substantial agreement on important concepts in Data Analysis and Data Science. Eighteen of the 25 most frequent concepts are shared by both fields. One difference is that the words *problem* and *solution* had Top 25 word rates for Data Science, but not for Data Analytics. A close look at Statistics concepts suggests that Data Analytics is more focused on *exploratory* concerns, such as searching for patterns in data. Data Science retains more of the classical *inferential* activities that use sample data to draw conclusions about populations. Both fields deal with Big Data situations, but Data Scientists must continue to be prepared for traditional small sample applications.

Keywords: data, data analytics, data science, statistics, exploratory, inference.

1. INTRODUCTION

Several decades ago, one of the authors worked as a Statistician for a food manufacturing company. Duties included research design, data collection, data management, and data analysis. The research design component involved relatively small laboratory experiments and sample surveys.

Data collection consisted of cleaning and organizing data onto punched cards or into a single flat file. For data analysis, existing software was used to perform analysis of variance (ANOVA), multiple regression, and cross-tabulation. Occasionally, small Fortran programs were written to perform custom data analyses, such as providing univariate and bivariate descriptive statistics, scatter diagrams, contour plots, and quality control charts.

For each research study, the typical size of the resulting data set was measured in kilobytes. Data management activities were minimal. If several related studies were performed, the generated samples were kept separate, without combining them into a single overall file.

Big Data

In the current era, many organizations now routinely collect massive amounts of data, both for individual studies and during continuing operations. Sizes of data sets for companies such as Walmart and Amazon are measured in gigabytes, terabytes, and beyond. A widely accepted term for these large data collections is Big Data.

With so much data being recorded by companies, organizations, and government entities, the skill-set for a Statistician, who must now deal with Big Data, requires considerable expansion. New approaches have been devised for the management and analysis of extremely large data sets.

Data Analytics and Data Science

Two recent fields that deal with Big Data have been developed and are evolving rapidly--*Data Analytics* (DA) and *Data Science* (DS). Universities are adding programs in these fields at the undergraduate and graduate levels. One Internet listing of 23 such academic programs includes 14 described as Data Analytics, 8 described as Data Science, and 1 program using both names.

A close inspection of these programs through reading academic descriptions and by examining required courses indicates many similarities but some notable differences. Generally, each program provides a mixture of Statistics, Applied Mathematics, Computer Science, and substantive content (e.g. Business, Medicine).

The program name Data Science suggests a careful application of the scientific method, especially research design, sampling, and measurement. The name Data Analytics places more emphasis on ways to describe large data sets. The central goal of both programs is to obtain practical interpretations of data that can assist in making operational and strategic decisions.

Purpose of this Research

In this study, we compare the topics and tools that are presented in Data Analytics and Data Science programs. Our research is relevant to potential students who need to evaluate the knowledge and skills provided in competing academic programs. It is also of value to faculty and academic administrators who are asked to design and teach courses in these programs.

Our research approach involves performing a *content analysis* (Krippendorff, 2012) of selected documents that describe these fields. Words used frequently in each sample of documents allow us to infer which concepts are emphasized in the two types of programs.

2. METHODOLOGY

This section describes the methodology used to collect word frequency data from samples of Data Analysis and Data Science documents. Special attention is focused on words that are relevant to Big Data issues.

Samples of Documents

Using the Internet, we collected a sample of 14 Data Analysis documents and a second sample of 12 Data Science documents that explain the nature of these fields. We chose documents that are available on the Internet and can be downloaded as PDF files (which are easily converted to text files). The size of the individual documents varied, but the total number of words in each sample was approximately the same.

Convert PDF files to Text Files

Documents in PDF file format are not convenient for performing repeated word searching and counting. Fortunately, Adobe Reader includes an option to convert the contents of a PDF file to a text file. We used Adobe Reader to create a text file for each of the 26 documents in our study.

Identify Individual Words in Documents

We observed that the document text files included many character strings that contain digits, punctuation, and other non-alphabetic symbols. They also contained a large number of common English words (e.g. "the", "and", "or") which were not of interest for this study.

To simplify our counting of concept words, we wrote a short Python program that performed the following "data cleansing" tasks.

(1) Our program first changed all letters to *lower-case*.

(2) The program then removed all *non-letter* symbols and replaced them with blanks.

(3) The program converted most *plural* nouns and verbs to *singular* form.

(4) Finally, our program removed approximately 120 common English words that appear on Fry's Lists (1993).

We used our Python program to obtain a filtered set of text files that ultimately consisted of lower-case letters and blanks, the singular form of nouns and verbs (but allowed different verb tenses), and excluded many common English words.

Perform Word Counts

We used a popular program called TextSTAT (Huning, 2007) to obtain word counts for all words in our "cleansed" text files. With TextStat, you first define a "Corpus" which holds a list of text files. We defined one corpus for the 14 DA files and a separate corpus for the 12 DS files.

To perform a word search, a separate TextSTAT screen allows the user to specify search options. Most of the time, we used the option to include all words, with the words and their counts presented in decreasing frequency order. We then went through the output and recorded word counts for the most frequent words.

Occasionally, we would enter a short string (e.g. *statistic*) to search for all words that contain the string (e.g. *statistic*, *statistical*, *statistician*).

Word Groups for Concepts

A single data-oriented concept can often be expressed by an author in more than one form. For example, nouns and verbs can be presented in singular or plural form. Verbs can also be written using various tenses. Sometimes, the same concept is described by both a noun and a verb (e.g. "sample", "sampling"). In some cases, *synonyms* representing similar ideas are used to represent a concept (e.g. "algorithm", "method"). Some concepts are written not as a single word but as a word phrase (e.g. "big data").

Our intent was to count how often authors referred to DA and DS concepts. However, TextSTAT was designed to count individual words. For this reason, we defined a *word group* for each concept. A word group is comprised of either a single word or a set of words that represent the same concept. To get a combined count for a concept, we added the frequencies for each of the words in the word group. This was the most time consuming part of our data collection and analysis.

Convert Word Counts to Word Rates

Because the DA and DS samples of documents contain different numbers of words, the actual word counts for a concept are not comparable across samples. To standardize the counts, we converted each word count for a concept to a *word rate*. The rate we chose was "per 100,000 words". Word rates were calculated for each concept in each set of documents.

3. ANALYSIS OF DATA

Our primary objective in this research is to examine the fields of Data Analytics and Data Science through the prism of selected documents which describe these fields. Which concepts do they share? In what ways are they different? To answer these questions, we compared word rates for concepts in several different ways, as is shown in the following tables.

Most Frequent Words

Table 1 provides listings of the 25 concepts with highest word rates for DA and DS. A separate list of concepts, ordered by decreasing word rate, is shown for each set of documents.

Eighteen concepts occur on both lists, but in different orderings. The most frequent word on both lists is *data*, which might be expected based on the names for the two fields. The second most frequent concept is *model*.

The DA and DS word rates for *data* are more than twice as high as the corresponding word rates for *model*, with word rates declining gradually for the remainder of the concepts in Table 1. Other concepts with high word rates on both lists include: *value*, *variable*, *function*, *set/element*, and *cluster*.

No.	DA Word	Rate	DS Word	Rate
1	data	1880	data	2517
2	model	753	model	845
3	point	572	**science**	654
4	value	565	algorithm/ method	523
5	mean/ average	555	set/element	521
6	function	543	value	512
7	variable	472	probability	458
8	**regression**	461	function	453
9	set/element	454	variable	438
10	cluster	439	cluster	422
11	matrix	386	**user/ customer**	416
12	distribution	383	point	413
13	algorithm/ method	380	**solution/ result**	399
14	**analytics**	364	mean/ average	380
15	**estimate**	334	number	378
16	node/vertex	319	**random**	351
17	**big**	318	node/vertex	349
18	number	317	vector	341
19	probability	308	edge/line	333
20	vector	307	matrix	322
21	**linear**	305	**statistic**	312
22	**sample**	295	**graph**	306
23	edge/line	290	**problem**	287
24	analysis	286	analysis	284
25	**tree**	280	distribution	265

Table 1: Top 25 Words - DA vs. DS
Words on a single list are in **bold**.

Seven concepts (shown in **bold**) are unique to each list. *Science* is high on the DS list, whereas *analytics* is relatively high on the DA list. This is not a surprise, and it attests in a minor way to the validity of the data.

Big (data), *regression*, *estimate*, and *sample* are among the Top 25 DA concepts. *Statistic*, *random*, *graph*, and *user/customer* are Top 25 DS concepts. Reasons for these differences are discussed later in the paper.

Because DA and DS are often described as interdisciplinary fields, we divided many of the concepts into separate tables according to four subject matter categories. Our groups include: Computational Mathematics, Statistics, Discrete Mathematics, and Software Development. These choices reflect our opinion that DA and DS adopt concepts to varying degrees from each of these fields. Some concepts are favored by DA, and others by DS. We added an extra table to present concepts that apply specifically to DA and/or DS (e.g. *analytics*, *science*).

Computational Mathematics

Some concepts apply to more than one field. For example, *analysis* can refer to an early stage in Software Development, or it can specify a particular Statistics methodology (e.g. analysis of variance).

In an earlier study (McMaster, 2007), we searched 56 Mathematics books for concepts that are common throughout Computational Mathematics. We examined textbooks representing Linear Algebra, Differential Equations, Discrete Mathematics, Statistics, Probability, and Operations Research. Our choice of Math fields was guided by the curriculum in the Applied and Computational Mathematics program at Princeton University.

We found 9 main concepts that are used broadly in Computational Mathematics. These concepts, along with their DA and DS word rates from the current study, are presented in Table 2.

No.	CM Word	DA Rate	DS Rate
1	model	**753**	**845**
2	value	**565**	**512**
3	function	**543**	**453**
4	algorithm/ method	**380**	**523**
5	variable	**472**	**438**
6	solution/result	229	**399**
7	problem	171	**287**
8	system	178	248
9	condition/ constraint	197	118

Table 2: Computational Math Words
Top 25 Word Rates are in **bold**.

The Computational Math concepts are listed in decreasing order by the larger of the DA and DS word rates. The high rates for these concepts indicate that DA and DS use these mathematical abstractions frequently to define and represent data.

A *variable* is an abstraction for a set of measurements (*values*) that become data. *Functions* and *models* describe patterns and relationships in data. *Algorithms* define calculations that can be used to identify a specific model for a data set.
The higher DS word rates for *problem* and *solution/result* suggest that DS pays more attention to *problem solving*. DA might be more

interested in finding data patterns to assist people in making decisions in a variety of situations, rather than in solving one particular problem.

A focus on problem solving in Mathematics books, even in books on Applied Mathematics, is not as common as one might expect. The majority of advanced Math books tend to organize material logically in a more familiar (to mathematicians) *theorem-proof* format. Polya's (1945) "How to Solve It" is one of the earliest and best known Math books having a clear emphasis on problem solving. This book is still in print and is highly regarded today.

By comparison, a more recent book on "How to Prove It" (Velleman, 1994) does not appeal to a wide audience. Perhaps this is because most Mathematics books are already based on the "how to prove it" framework.

Statistics
A list of 18 Statistics concepts, many with high word rates, is given in Table 3. This table does not include Computational Math concepts presented in Table 2, even though some of these concepts apply to Statistics. As in the previous table, the concepts are listed in decreasing order by the larger (DA or DS) word rate.

No.	ST Word	DA Rate	DS Rate
1	data	**1880**	**2517**
2	mean/average	**555**	**380**
3	regression	**461**	179
4	probability	**308**	**458**
5	cluster	**439**	**422**
6	distribution	**383**	**265**
7	random	168	**351**
8	estimate	**334**	126
9	statistic	233	**312**
10	sample	**295**	157
11	analysis	**286**	**284**
12	test	264	203
13	predict	254	223
14	error	251	154
15	plot	228	107
16	variance	221	91
17	component	198	133
18	density	198	68

Table 3: Statistics Words
Top 25 Word Rates are in **bold**.

In Table 3, 9 DA concepts and 8 DS concepts have Top 25 word rates. The word having the highest rate on both lists is *data*. We consider *data* primarily as a Statistics concept, even though it is used frequently in computing and applied fields (e.g. science, business, government). From our previous research, we found that the word *data* appears infrequently in most Math books, including Applied Math books.

The 6 concepts having Top 25 word rates for both DA and DS are: *data*, *mean/average*, *probability*, *cluster*, *distribution*, and *analysis*. *Regression*, *estimate*, and *sample* are Top 25 concepts for DA. *Random* and *statistic* are Top 25 concepts for DS.

The field of statistics can be divided into *exploratory* and *inferential* activities. Exploratory methods search for patterns in the sample data, with less regard to the source of the data and the manner of sampling. Inferential statistics uses sample data to evaluate claims (hypotheses) about the population from which the sample was drawn.

Inferential statistics requires *probability* models based on how the data is collected. Usually, the basis is random sampling in surveys or randomization in experiments. Observe in Table 3 that the word rates for *probability* and *random* are higher for DS than for DA, since DS focuses more on inference.

On the other hand, the word rates for *regression*, *mean/average*, *estimate*, and *sample* are higher for DA. These concepts describe characteristics of the sample data.

Discrete Mathematics
Discrete Mathematics is a topic taught to Mathematics students and Computer Science students. In the CS curriculum, the course is often called Discrete Structures. Table 4 lists 12 Discrete Math concepts, along with their DA and DS word rates. Again, the concepts are listed in decreasing order by the larger (DA or DS) rate.

Discrete Math models are consistent with the discrete nature of data in a computer. Continuous Math models such as differential equations require floating point numbers and careful computation techniques employing numerical methods.

The word rates for DA and DS are surprisingly similar. Nine of the concepts are Top 25 DA words. Eight of the concepts are Top 25 DS words. The first 7 concepts on the list are Top 25 words for both fields.

No.	DM Word	DA Rate	DS Rate
1	point	**572**	**413**
2	set/element	**454**	**521**
3	edge/line	**290**	**333**
4	matrix	**386**	**322**
5	number	**317**	**378**
6	node/vertex	**319**	**349**
7	vector	**307**	**341**
8	graph	243	**306**
9	linear	**305**	172
10	tree	**280**	150
11	dimension	191	237
12	distance	133	215

Table 4: Discrete Math Words
Top 25 Word Rates are in **bold**.

Most of the word rate differences are relatively small. The largest differences are for *point*, *linear*, and *tree*, with DA having the higher rates.

Most of the Discrete Math concepts on the list define data structures (*matrix*, *vector*, *point*), finite models for data (*graph*, *tree*), and special features of the models (*node/vertex*, *edge/line*, *dimension*, *distance*). These models and data structures tend to be applied to sample data patterns, rather than to draw inferences to populations.

Software Development
Table 5 lists 10 concepts that relate to the creation of software. We call this process Software Development.

No.	SE Word	DA Rate	DS Rate
1	user/customer	216	**416**
2	class/object	275	182
3	case	215	165
4	input/output	198	204
5	code/software	194	183
6	table	138	161
7	type	126	158
8	attribute	143	53
9	database	134	121
10	file	91	109

Table 5: Software Development Words
Top 25 Word Rates are in **bold**.

Software Development allows us to see data and algorithms from a computer's point of view, which can improve our understanding of DA and DS. The value of computers in DA and DS is not limited to the ability of computers to transform data rapidly. Practitioners also benefit when they are able to translate data structures and algorithms into a language that is understandable to the computer (Knuth, 2008).

In Tables 2 thru 4, over half of the concepts have Top 25 word rates for DA and DS. In Table 5, the only Software Development concept that has a Top 25 word rate is *user/customer* for DS. The user/customer usually provides the problem for the Data Scientist or Data Analyst to solve, plus a request for software.

The concept *class/object* barely misses having a Top 25 word rate for DA). This is a foundation concept for developing software components using object-oriented programming (OOP). The remaining concepts in Table 5 have word rates above 100 for DA and/or DS. These topics are discussed in our sample documents, but at a lower rate than most concepts in earlier tables. This indicates that these Software Development concepts are relevant to DA and DS, but do not receive the same level of coverage by our sample of authors.

Most DA and DS academic programs require at least one programming course. However, the amount of programming that is required of students in a "non-programming" course can vary greatly.

We note that the concepts *database*, *table*, and *attribute* have fairly low word rates in Table 5. Their low word rates do not diminish the importance of database principles and software for managing Big Data.

Data Analytics and Data Science
In Table 6, we highlight 9 concepts that are important to DA or DS, but do not fit well into any of our previous categories.

No.	SE Word	DA Rate	DS Rate
1	science	58	**654**
2	analytics	**364**	141
3	big (data)	**318**	90
4	learning	97	237
5	visualization	29	190
6	training	145	187
7	hadoop	143	31
8	machine	59	134
9	mining	134	21

Table 6: Data Analytics and Data Science
Top 25 Word Rates are in **bold**.

We can think of these words as DA-specific or DS-specific. We have already mentioned the Top 25 word rates for *analytics* (DA) and *science* (DS).

The word *big* (usually stated as *big data*) has a Top 25 word rate only for DA.

The remaining 6 concepts describe models and methods for DA or DS. Four of the concepts have higher DS rates (*learning*, *visualization*, *training*, and *machine*). Two of the concepts have higher DA rates (*hadoop* and *mining*).

Machine and learning are usually expressed as the single concept *machine learning*. Visualization and mining are usually combined with the word data, as in *data visualization* and *data mining*. Hadoop is widely-used open source software for the management and parallel processing of big data.

The bottom 6 concepts in Table 6 apply in varying degrees to both DA and DS. The differences in their low word rates could partially be due to our small samples of documents.

4. CONCLUSIONS

In this research study, we compared word rates for concepts mentioned most often in samples of Data Analytics and Data Science documents. Our analysis of word rates leads us to the following conclusions.

First, there is substantial agreement on the most important concepts in DA and DS. The 25 most frequent concepts in each field share 18 of these concepts.

The most frequent concept in both fields is *data*. The word rate for data is more than twice the second highest rate, which is for *model*. Given the "D" in the names of the DA and DS fields, the frequent mention of data is not surprising. However, in earlier research (McMaster, 2007) we found that books on Mathematics topics often favor a logical framework (*theorem*, *proof*) over an empirical approach (*data*). You can think of DA and DS as leading a renaissance for data.

Second, when the concepts in our documents were grouped into the categories of Computational Math, Statistics, and Discrete Math, the concepts with highest rates tended to be the same for DA and DS.

In the Computational Math category, *variable*, *value*, *model*, *function*, and *algorithm/method* had high rates for both DA and DS, but *problem* and *solution/result* had noticeably lower rates for DA. This suggests that DS places more emphasis on *problem solving*.

We included a category for Software Development concepts, since DA and DS can be viewed as a blend of Statistics, Mathematics, and Computer Science. Almost all of the Software Development concepts had low word rates in the DA and DS documents. The explanation for low word rates might be partially due to the choice of documents in our samples. On the other hand, data analysts and data scientists do not write most of the software they use, so less emphasis on programming is understandable.

However, three of the Software Development concepts with low word rates--*database*, *table*, and *attribute*--are only indirectly involved in writing code. DA and DS without databases would be ineffective, so the lack of discussion about databases is disappointing.

Third, a closer look at Statistics concepts with differing DA and DS word rates suggests that DA places more focus on *exploratory* concerns, such as searching for patterns in sample data. DS retains more of the classical *inferential* activities that use sample data to draw conclusions about populations.

One reason that DS retains more focus on inferential statistics is due to sample size considerations. Both DA and DS often deal with Big Data situations. DA has a higher word rate for *big* (data), but Data Scientists must also be prepared for traditional small sample problems.

Inferential statistics requires probability models based on the data collection methodology. The probability distribution for a statistic (*sampling distribution*) varies with the sample size. In many cases, the variance of the statistic is inversely proportional to the sample size. An extremely large sample size will yield a very small variance for the statistic. When the sample size is large, a "significant" (but small) difference in the sample may be unimportant for practical reasons. Thus, in Big Data cases, the sample can be considered to be the entire population, making inference irrelevant.

Future Research

Future research is planned for the following Big Data studies:

1. Repeat this study with larger and more representative samples of documents. The literature on Data Analytics and Data Science is growing rapidly. In addition, the fields themselves are evolving in goals, methods, and applications.

2. Perform a comparison of program outlines and course descriptions for the ever-increasing number of graduate and undergraduate degrees offered in Data Analytics and Data Science. We would record which courses form the core of the programs and which topics are available as electives.

3. Perform an analysis of several Big Data projects to examine what types of applications are represented, what methodologies they employ, and how they measure "success".

5. REFERENCES

Fry, E.G., Kress, J. E., & Fountoukidis, D.L. (1993), *The Reading Teacher's Book of Lists* (3rd ed). Center for Applied Research in Education.

Huning, M.*TextSTAT 2.7 User’s Guide.TextSTAT*, created by Gena Bennett, 2007.

Knuth, D. (2008), "Donald Knuth: A Life's Work Interrupted." Communications of the ACM, Vol. 51, No. 8.

Krippendorff, K. H.*Content Analysis: An Introduction to Its Methodology*, 3rd Ed. SAGE Publications, 2012.

McMaster, K., Hadfield, S., Wolthuis, S., & Sambasivam, S. (2007), "Two Gestalts for Mathematics: Logical vs. Computational." Proceedings of ISECON 2007, Vol. 24.

Polya, G. (1945). *How To Solve It*. Princeton University Press.

Velleman, D. (1994). *How to Prove It: A Structured Approach*. Cambridge University Press.

Data Analytics documents (partial list)

Hadoop: Big Data Analysis Framework.
Tutorials Point, 2014.

Ledolter, J., *Data Mining and Business Analytics With R*. John Wiley & Sons, 2013.

Shalizi, C. R., *Advanced Data Analysis from an Elementary Point of View*. Spring 2013.

Wesler, M., *Big Data Analytics For Dummies*. John Wiley & Sons, 2013.

Zaki, M., & Wagner M., *Data Mining and Analysis*. Cambridge University Press, 2014.

Data Science documents (partial list)

Grus, J., Data Science from Scratch. O'Reilly Media, 2015.

Hopcroft, J., &Kannan, R.*Foundations of Data Science*, Draft, 2014.

Herman, M., Rivera, S., Mills, S., Sullivan, J., Guerra, P., Cosmas, A., Farris, D., Kohlwey, E., Yacci, P., Keller, B., Kherlopian, A., & Kim, M.*The Field Guide to Data Science*. Booz, Allen, Hamilton, 2013.

Pierson, L.*Data Science For Dummies*. John Wiley & Sons, 2015.

Zumel, N., & John M., *Practical Data Science with R*. Manning Publications, 2014.

The Challenges of Teaching Business Analytics: Finding Real Big Data for Business Students

Alexander Y. Yap
ayyap@ncat.edu

Sherrie L. Drye
sldrye@ncat.edu

Department of Business Education
North Carolina A&T State University
Greensboro, North Carolina 27411, USA

Abstract

This research shares the challenges of bringing in real-world big business data into the classroom so students can experience how today's business decisions can improve with the strategic use of data analytics. Finding a true big data set that provides real world business transactions and operational data has been a challenge for academics developing a data analytics course or curriculum, because in the past academics use to rely on 'fictitious small data' to teach students the basics of analytics. The ideal scenario for business academics who wish to bring a more cutting-edge experience to business students is to show them how evolutionary tools in data mining analytics can interpret real world big business data. This research emphasizes the need for students to have more exposure to big data analytics. This paper presents how a real world big data set has been utilized in a business course.

Keywords: Education in Data Analytics, Real World Big Data, Data Warehousing, Data Mining, Business Analytics

1. INTRODUCTION

Many business schools have shown interest in offering courses in Big Data Analytics due to the significant demand of companies for data scientists (Davenport & Patil, 2012; O'Connor, 2012). There is a severe shortage of skilled data scientists graduating into the workforce, with reports stating that by 2018, the supply will exceed the demand by anywhere from 200,000 to over a million jobs (Brown, 2016; Overly, 2013). Employers readily admit that they need "numerate employees" with "quantitative aptitude," "data literacy" skills, and a "data-driven mindset" (Overly, 2013, p. 2; Harris, 2012, p. 2). Most employees will be working with big data in some way. Some will need primarily skills and insights to interpret the results of data analysis. These are called citizen data scientists by Gartner (Marr, How The Citizen Data Scientist Will Democratize Big Data, 2016). Other jobs will entail a deeper knowledge of statistics, programming, data cleansing, data management, visualization, and data analysis techniques (Thibodeau, 2012).
Companies often advertise information technology and related jobs based on their "ideal" qualifications, with so many qualifications listed that no person really has all of the skills advertised. Usually they are able to choose the closest candidate to what the ideal would be.

However, big data jobs are different for a number of reasons. First, job descriptions for big data jobs are difficult to write since it is such a new field (Thibodeau, 2012). What skills do you include when the job description is unclear? Companies need to examine what their needs are and not seek the traditionally "ideal" candidate. For big data jobs, they cannot necessarily be entrenched in requiring knowledge of certain technology as they could be with typical information technology jobs (Brown, 2016). Second, because universities are not graduating enough students with data science skills, companies are often training them in-house (Tschakert, Kokina, Koziowski, & Vasarhelyi, 2016; Brown, 2016). The shortage of university graduates is critical. The state of Illinois actually set up an alliance, IoT Talent Consortium (Internet of Things Talent Consortium) with Microsoft to increase the number of skilled workers in the state. They set up an online certification program (www.iottalent.org/illinois-datascience) where any resident (and up to 100 prisoners in the state penal system) could take nine courses in "analytics, predictive analytics, various coding languages, and other digital disciplines" (Shueh, 2017, p. 1). Lastly, it may not be a matter of choosing one ideal person, but matching people with varying skills into teams to overcome the skills gap and to address the complexity of doing big data well (Brown, 2016).

The skills gap is real and it is in the face of companies and universities. Although universities are struggling to meet the demand, they are falling short of meeting the needs of the job market. One reason is that they may not have the technology and resources to teach big data. It is a good idea to start early with statistical analyses, Excel, and Access, but bigger tools and data are needed (Tschakert, Kokina, Koziowski, & Vasarhelyi, 2016; Thibodeau, 2012). There are analytic tools such as R, which are free and readily available (Bisson, 2017), but there are only a few offered and used in universities. Even if R is free, the trick is how to teach the analytical and interpretive side of the big data analysis. IBM released the Watson and Other Natural Language Processing platform for widely available use in universities and even has a student version of its Watson Analytics platform (Marr, How IBM is Hoping to Close the Massive Big Data and Analytics Skills Gap, 2016). The natural language processing can skip the statistical middleman so that users do not have to know all the statistical techniques and can go straight to interpretation after machine analysis (Marr, How IBM is Hoping to Close the Massive Big Data and Analytics Skills Gap, 2016). What academia needs is more tools and big data to teach these critical skills.

2. THE SEARCH FOR BIG BUSINESS DATA IN ACADEME

At academic conferences and seminars on "big data", the demo data or the sample data in these academic gatherings ironically end up as "small data" samples. The noble intent is to showcase some academic exercises which can be used in the classroom, but the data is not "big" by any means. Software vendors who partnered with academia so their big data analytics software can be employed in Big Data Business Analytics classes have not really offered "big" business data samples, either. What they offer are good small data samples with probably up to a few hundred entries that would give students an appreciation of organizational data, but the data sample were not true samples of real world big data.

It appears that real world business organizations have not been that willing to publicly share their real big data samples to the academic community due to fear of privacy and fear that their competition will have access to publicly-shared data. At a conference attended by one of the authors, a lecturer, who admitted frustration for not getting access to big business data, suggested that big data academicians in business schools can use the free weather big data that meteorologists, oceanographers, and climate scientists use to predict the weather using predictive analytics. However, as instructors of big data analytics classes, we want to find real big business data that is relevant for business decisions instead of predicting the weather. Business students need to see big business data and not big weather data.

3. OBJECTIVE OF RESEARCH

This objective of this case study paper is threefold: (1) To share our experience why using real world big business data can pose a challenge for business academics teaching big data. (2) To share our quest to offer students a real big data set in a big data course curriculum. Where can we find real big business data that can be used in the classroom to teach big data business analytics? (3) Lastly, we want to showcase the software tools we used to teach big data analytics.

4. THE CASE OF SAM'S CLUB BIG DATA

Sam's Club has donated its big data set to Sam Walton's College of the University of Arkansas. The data is more than a few years old but it has transaction data dating back to the 1990s. While it is not the most current set of data, it is a real Big Data warehouse that contains millions of historic data entry and captured detailed retail transactions and store information of Sam's Club across the United States. This big data can be used to teach Big Data Analytics specifically for business students.

As a member of the academic community, an instructor can apply for access to the big data set with the University of Arkansas (https://walton.uark.edu/enterprise/). The *Teradata University Network* (TUN) is the organization managing the partnership between different technology providers and access to real word big business data. The big data set is stored in a Teradata warehouse with 20 terabytes of storage and with 576 Gigabytes of RAM memory. Teradata is an independent solutions provider of Data warehouse, Data mining, and Big Data analytics solutions. Teradata provide big data solutions to companies like PayPal, Proctor and Gamble, Wells Fargo, T-Mobile, and other well-known Fortune 500 companies. The biggest data set in TUN is from Sam's Club, but the TUN network also has big data for other retailers, like Dillards, for example. Other software vendors like SAP, SAS, and Microsoft software platforms can access the Data warehouse through universal ODBC connections, given the correct password credentials. Figure 1 shows the connections in a network diagram.

Challenges

There are several challenges to instructors in using this system. These are discussed based on our experience of teaching this system for the past two years.

(1) **Connecting to the TUN Network** - Sam's Club/Walmart has an agreement with the Sam Walton's College of Business in the University of Arkansas that they will donate the big data set for academic use, but it cannot be taken out of the University's system. So, if you are an instructor in another university, you and your students can only access the big business data via Remote Desktop connection to the TUN network (See Figure 2). Occasionally, this poses a problem for laboratory classroom computers that do not have a huge amount of RAM memory. And if the lab classrooms are running on VMWare clients connected to a virtual machine (VM) server, the administrators should also allocate an adequate amount of RAM memory for the lab computers clients functioning as virtual machines. Our experience is that with limited RAM memory in VM clients, remote desktop can be slow and can cause the VM clients to reboot. In fact, one University administrator suggested that the VM clients need to be rebooted if the memory cache is full. Students complain that their lab exercises have been interrupted by these freezing and reboot problems. However, once rebooted, the access and connection to the University of Arkansas TUNS server tends to be more stable. So if the class being taught is after 2:00 pm and the laboratory computers have been used by other classes before that, it is best to reboot the laboratory computers before all students go into their remote desktop sessions.

In addition to the challenges of using remote desktop on laboratory computers that may not cooperate with students, the other challenge falls on the tenacity of students to set up the remote desktop connection. It is normal to see a couple of students who read the instructions quickly and cannot get any remote desktop connection for more than 30 minutes into the laboratory exercises, and they get frustrated. Most of the time, it has to do with spelling the server information wrong. It is common to see students type the server address wrong like "waltoncollege.urak.edu" instead of "waltoncollege.uark.edu" and they think it is the TUN server that is not letting them in. Sometimes it does take up to 7-15 minutes for the TUN server to negotiate and recognize the username and password credentials when it is used for the very first time. When everything does not work out, it is best to give the student an alternative username and password. As instructors, it is wise to ask the TUN administrator for more usernames and passwords than the number of students in the class, so if there is a non-working username/password you have extra ones on hand. Based on our experience, this rarely happens, but the extra credentials come in handy.

(2) **The Lack of Metadata and Data Content descriptions** - For instructors wishing to learn about the big data content, the negative aspect is the lack of metadata descriptions about the data fields. There are a few metadata description samples, but does not cover everything in the Big Data set. The administrators of the TUNs server and data warehouse system have done an excellent job making sure that the data warehouse and all the Teradata software are running in excellent condition. But since they are not employees of Sams' Club, they cannot answer questions about the data content itself or what some of the data field names stand for. Therefore, calling them for assistance is not very helpful at times. Instructors wishing to use this big data need to explore the content and determine what the data content is all about, because the metadata descriptions are mostly lacking from the brief manual and instructions.

In Figure 3, when one opens the *"item_desc"* table and one can see several data fields like *Item_Nbr, Category_Nbr, Sub_Category_Nbr, Primary_Desc, and UPC*, a person without any business background would probably not immediately recognize these data field syntaxes. However, if a user is familiar with a Product table, these fields begin to look familiar. Most business organizations selling physical artifacts will have a data table containing all information about their products, including description, color, weight, price, and vendor information. Once we extracted the data from this table, we confirmed that the "item_desc" table is fundamentally a Product table that contains information about various products that Sams Club sells. Item_Nbr is essentially the unique Product identification number. And since each product belong to a certain product category, like "flat screen television" belongs to the product category "Electronics", Sam's Club assigns a Product Category identification number for each Product category. There is also a Product sub-category, like "Televisions and Video Products", under the larger product category of "Electronics". So the data set also has a Product Sub-Category identification number. The data field "Primary_Desc" actually contains the name of product, hence the wording 'primary description'. There is also another data field called "Secondary_Desc", or secondary description. This field provides additional or more detailed product description, in addition, to the product name. If the product has a color and size, the data fields "color_desc" and "size_desc" will provide information about that. And the UPC data field is the "universal product code (UPC)" that we find in 'bar codes' and which the barcode scanners pick up.

For instructors who wish to use this big data set, it is a plus if the instructor has some familiarity with business data, because the data field descriptions are not completely provided in the sparse 'user's manual' and you may have to spend some time going over the big data content and understanding what the data field names stand for and what contents are inside these data fields.

(3) **Learning SQL** – A common question from both business students and faculty inquiring about teaching this big data system is whether there are some programming skills involved. The answer is that students will need a basic understand of the SQL language and how data sets are created, structured, and connected to other data sets in order to extract data from the data warehouse.

In Figures 3, using the basic SQL "select" and "from" allows one to determine how many Sam's Club members there are and how many unique products Sam's Club carry in this data set. Running the SQL script, we found that the big data set has recorded a total of about 5.66 million Sam's club members and more than 432,000 unique product items, as the reference for all retail transactions in this data set.

Having previously taught 'Introduction to Database' classes, students in those classes are shown sample data sets that had less than a hundred items or records. With a true big data class, students slowly begin to realize that we are going through hundreds of thousands of items to millions of data records. Occasionally, extracting and filtering such a huge data set could take 20 – 30 minutes to execute, before it displays the results. Students often ask "Why is it still not finished? The query has been already running for 20 minutes? "

More complex SQL scripts can answer questions like 'What is the average sales of electronic items for Sam's Club stores located in Wisconsin from January 1st to January 30th ' The more one 'slices and dices' the data set, the more SQL skills one needs to have using the Teradata system. In teaching this course, we need to give students some sample SQL scripts when we try to extract a specific data set in order for them to appreciate (1) how data is extracted to answer specific questions; (2) what type of data set they are extracting; and (3) how that is relevant to making certain business decisions if one were to assume the role of a Sam's Club manager who decides how much inventory to carry, what is the best price mark-up for each item, and what colors and sizes are more demanded by customers, and what coupons to issue for various products. Using big data analytics can help business manager with such decisions.

Most business students do not have a background in SQL, nor are they required to take SQL in AACSB business degrees. So, in a business school environment where SQL is not a required course, there is a need to start introducing students to basic SQL scripting for extracting data from a Teradata warehouse. Figure 6 shows a sample of how to extract data using a SQL query from the big data set to display product information.

(4) **Big Data Analytics Software** - Teradata has two main software packages that work with the Data warehouse. The first is the Teradata SQL Assistant that allows users to write

SQL scripts in order to extract specific information from Sam's Club big data (Figure 6). The other software is the Teradata Analytics Miner, which can perform several analytic techniques that run on intelligent algorithms that search for unique data patterns. The challenge of teaching students analytics is to put these statistical tools in the perspective of making better management decisions with these data and tools.

Teaching Analytics Techniques

Some examples of analytics techniques we use for this course are as follows:

Cluster Analysis

This analytics tool allows the users to search for spending patterns of Sam's Club members. For example, one could search for "how much does each customer spend every time they visit a Sam's Club store?" The dataset we look at is called "Total Visit Amount". In Figure 4, we then ask the analytics software to look for 5 distinct clusters and we also ask the clustering algorithm to go through the entire big data set in 25 iteration cycles. This means that the algorithm combs through the big data set 25 times to find 5 cluster data patterns. The more iteration the algorithm does, the more it will accurately detect and identify cluster patterns. Of course, we can ask the software to look for 4 or 6 clusters, instead of 5 clusters, and we could ask it to perform 50 iterations. It is mostly a trial and error process to determine how many clusters will best represent the data analysis we are looking for.

The result (Figure 5) shows that the clustering algorithm has identified five clusters of consumer spending. The first cluster is $95.32 and 77% of Sam's Club customers spend that average amount of money every time they visit a Sam's Club store. The second cluster is $255.97 and 20% of customers spend that average amount of money every store visit. The third cluster shows that about 3% of Sam's Club customers spend $552.49 every time they visit the store. And there are two other clusters that are probably a fraction of 1%, namely $2,811.18 and $94,356.48. These are large amounts because Sam's Club has corporate members. And corporations spend more than individual consumers.

It is important for students to know that a large percentage of Sam's Club members (77%) spend no more than $100 every time they visit a store, so managers need to make a decision what kind of products and brand they carry to fit that type of shopper. And there are 20% of members that spend about $256 every time they visit a store, and they may have preferences for high end product brands and their spending level needs to be met by carrying certain high-end quality products. As a side note, these dollar numbers are from a decade ago at the time of this writing, so in today's dollar value – these dollar amounts could be 30%-50% higher.

Association Analytics

This analytics tool can search through big data sets generated by each Sam's Club store and see if there is any unique consumption pattern linking two products. For example, a large majority of people who buy hamburgers in grocery stores may also buy hotdogs. Consumers who buy paper towels may also buy toilet paper. This is called associative analytics, which many retailers use to determine which related products consumers may be interested in. Many retailers issue coupons to customers who may be interested in products that are highly associated with other products in terms of consumption habits. Most Dads who pick up a six-pack of beer could also pick up diapers for their kids, and the next time they get beer they could get a $2 discount coupon for diapers.

In Figure 6, the Associative Analytics software of Teradata allows users to choose the Product Name contained in the data field "Primary_Desc" and the number of times people buy products every time they visit a Sam's Club store (Visit_Nbr). The analytics algorithm tries to find one-on-one patterns between one product and another product. Would people buy Product A and then also buy Product B? How strong is the association between Product A and B? This is what the association tool wants to find out.

In Figure 7, we show an example of how at one particular Sam's Club store, people who buy 'Bounty Paper Towels' also buy 'Charmin 24 Double Roll Toilet papers' and this comes out with a high association showed at a ZScore of almost 68 (red color chart). We can also see in this chart that people who buy Extra large 18 pieces of eggs also buy 2% Fat Milk and 1% Fat Milk (green and blue color).

Decision Tree Analytics

A third popular analytics tool is the decision tree. Using this tool, we want to determine why consumers continue to become active or inactive members of Sam's Club. The data field "Member_Status_CD" shows whether a member has an 'active', 'deactivated' or 'inactive', or 'expired' status. The dataset has a code (A, D, and E). Deactivated members are members who have been inactive for a long time. Their information is never deleted. The expired membership status is for members whose membership was just recently expired (maybe a few days or weeks) and they could renew and become active members again. In this exercise, students try to find out what causes members to stay 'active' with Sam's Club or stay 'inactive' and not renew their membership. Is it because they like the products being sold (Item_nbr), is it because they like to buy a certain quantity of items every time they visit (Item_Quantity), membership type (retail or corporate accounts), the different product categories available (category_nbr), the product sub-categories (Sub_category_desc) available, the location (store zip code), what store they registered as members (Register_Nbr), how much they spend or the amount scanned at the counter (Total_Scan_Amount) and what type of payment they use (Tender_type) which could be cash or credit card.

The results (Figure 8) show that the product categories (Category_Nbr), the products items being sold at the store (Item_Nbr), member type - retail and corporate memberships (member_type), the location of store (store_zip), the membership number which identifies location of member registration (Membership_Nbr), the payment method (Tender type), the amount spent by member (Total Scan Amount), are significant factors that determines a member's decision to be 'active' members or 'inactive members'. The analytics software also has a graphic representation of the decision tree, depicting how one variable affects the decision of people to be an active or inactive member.

5. STUDENT FEEDBACK

Student feedback about this analytics course has been very positive. For example, students rated "the practical application of the course" at 4.81 out of a 5.0 perfect score. In comparison, the average rating for this criterion for the entire University is at 4.22 and, for the Business College, the average is at 4.29. The comments were very positive. One student, for example, said, "I enjoyed the opportunity to work with the different data technologies used in the class. I was exposed to new programs that I feel will be very helpful to my career."

6. SUMMARY

The paper shares our experience in finding and teaching big data analytics in the hope that other business school instructors who are interested in the fast evolving field of big data analytics are able to get some insights into our experience. While we have met some challenges in teaching real world big data, the advantages for student getting exposure to real world big data sets, data warehouse systems, and data mining analytics prepares their mindset for the world of business where big data is becoming an invaluable asset for strategic decision making, planning, and forecasting.

While the availability of real world big business data for academic consumption is still limited, the availability of Sam's Club big data set is a good start for the academe to get a feel of what big business data really looks and feels like. Hopefully, more business organizations in the future can share their old big business data to the academic community and allow students to discover how they make more accurate business decision by performing data mining analytics on big data sets, and how they align their business goals in response to discoveries made by big data analytics.

7. REFERENCES

Bisson, S. (2017, January 11). *Microsoft's R Tools Bring Data Science to the Masses*. Retrieved May 26, 2017, from InfoWorld: http://www.infoworld.com/article/3156544/big-data/microsofts-r-tools-bring-data-science-to-the-masses.html

Brown, M. S. (2016, June 27). What Analytics Talent Shortage? How to Get and Keep the Talent You Need. *Forbes*. Retrieved May 26, 2017, from https://www.forbes.com/sites/metabrown/2016/06/27/what-analytics-talent-shortage-how-to-get-and-keep-the-talent-you-need/#5df4ab3018da

Davenport, T., & Patil, D. (2012). Data Scientist: The Sexiest Job of the 21st Century. *Harvard Business Review*. Retrieved May 27, 2017

Harris, J. (2012, September 13). Data is useless without the Skills to Analyze It. *Harvard Business Review*. Retrieved May 26, 2017, from https://hbr.org/2012/09/data-is-useless-without-the-skills

Marr, B. (2016, February 29). How IBM is Hoping to Close the Massive Big Data and Analytics Skills Gap. *Forbes*. Retrieved May 26, 2017, from https://www.forbes.com/sites/bernardmarr/2016/02/29/how-ibm-is-hoping-to-close-the-massive-big-data-and-analytics-skills-gap/#13f2ea14df87

Marr, B. (2016, April 1). *How The Citizen Data Scientist Will Democratize Big Data*. Retrieved from Forbes: https://www.forbes.com/sites/bernardmarr/2016/04/01/how-the-citizen-data-scientist-will-democratize-big-data/#6bb5a9d365b8

O'Connor, F. (2012, October 10). *Colleges Incorporate Data Science into Curriculums*. Retrieved May 26, 2017, from Computer World: http://www.computerworld.com/article/2492245/it-careers/colleges-incorporate-data-science-into-curriculums.html

Overly, S. (2013, September 15). As Demand for Big Data Analysts Grows, Schools Rush to Graduate Students with Necessary skills. *The Washington Post*. Retrieved May 26, 2017, from https://www.washingtonpost.com/business/capitalbusiness/as-demand-for-big-data-analysts-grows-schools-rush-to-graduate-students-with-necessary-skills/2013/09/13/afbafb3e-1a66-11e3-82ef-a059e54c49d0_story.html?utm_term=.62ad6830497d

Shueh, J. (2017, April 20). *Illinois Fights Technical Skills Gap with Courses in Data Science, Analytics, Coding*. Retrieved May 26, 2017, from Daily Scoop: http://statescoop.com/illinois-fights-technical-skills-gap-with-courses-in-data-science-analytics-coding

Thibodeau, P. (2012, October 24). *Q&A: What's Needed to Get a Big Data Job?* Retrieved May 26, 2017, from ComputerWorld: http://www.computerworld.com/article/2492880/big-data/q-a--what-s-needed-to-get-a-big-data-job-.html

Tschakert, N., Kokina, J., Koziowski, S., & Vasarhelyi, M. (2016). The Next Frontier in Data Analytics. *Journal of Accountancy*. Retrieved May 26, 2017, from http://www.journalofaccountancy.com/issues/2016/aug/data-analytics-skills.html

Editor's Note:

This paper was selected for inclusion in the journal as an EDSIGCON 2017 Distinguished Paper. The acceptance rate is typically 7% for this category of paper based on blind reviews from six or more peers including three or more former best papers authors who did not submit a paper in 2017.

2. APPENDIX 1 (FIGURES)

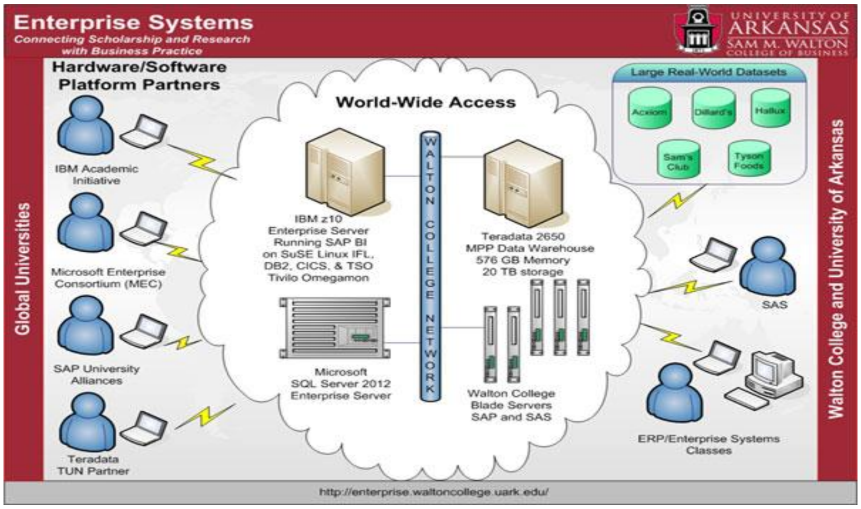
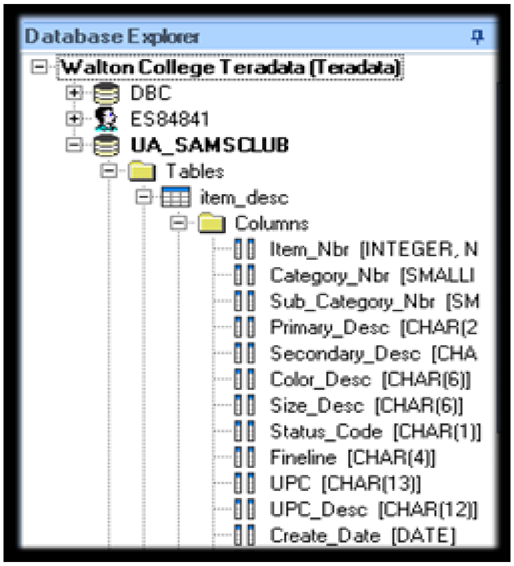
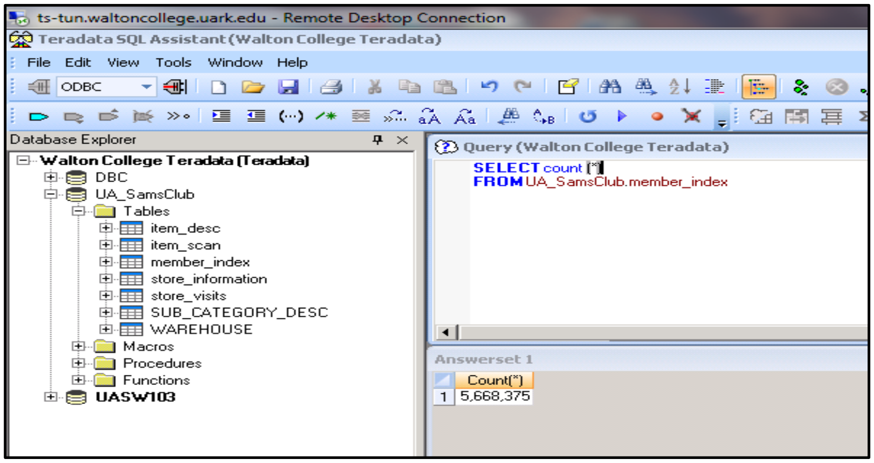

Enterprise Systems
Connecting Scholarship and Research with Business Practice
UNIVERSITY OF ARKANSAS SAM M. WALTON COLLEGE OF BUSINESS
Hardware/Software Platform Partners
IBM Academic Initiative
Microsoft Enterprise Consortium (MEC)
SAP University Alliances
Teradata TUN Partner
Global Universities
World-Wide Access
IBM z10 Enterprise Server Running SAP BI on SuSE Linux IFL, DB2, CICS, & TSO Tivilo Omegamon
Microsoft SQL Server 2012 Enterprise Server
WALTON COLLEGE NETWORK
Teradata 2650 MPP Data Warehouse 576 GB Memory 20 TB storage
Walton College Blade Servers SAP and SAS
Large Real-World Datasets
Acxiom
Dillard's
Hallux
Sam's Club
Tyson Foods
SAS
ERP/Enterprise Systems Classes
Walton College and University of Arkansas
http://enterprise.waltoncollege.uark.edu/

Figure 1. Connecting Universities to real world large datasets



```
Database Explorer
⊟ Walton College Teradata (Teradata)
    ⊞ DBC
    ⊞ ES84841
    ⊟ UA_SAMSCLUB
        ⊟ Tables
            ⊟ item_desc
                ⊟ Columns
                    Item_Nbr (INTEGER, N
                    Category_Nbr (SMALLI
                    Sub_Category_Nbr (SM
                    Primary_Desc (CHAR(2
                    Secondary_Desc (CHA
                    Color_Desc (CHAR(6))
                    Size_Desc (CHAR(6))
                    Status_Code (CHAR(1))
                    Fineline (CHAR(4))
                    UPC (CHAR(13))
                    UPC_Desc (CHAR(12))
                    Create_Date (DATE)
```

Figure 2. Deciphering the data field descriptions


ts-tun.waltoncollege.uark.edu - Remote Desktop Connection
Teradata SQL Assistant (Walton College Teradata)
File Edit View Tools Window Help
ODBC

```
Database Explorer
⊟ Walton College Teradata (Teradata)
    ⊞ DBC
    ⊟ UA_SamsClub
        ⊟ Tables
            ⊞ item_desc
            ⊞ item_scan
            ⊞ member_index
            ⊞ store_information
            ⊞ store_visits
            ⊞ SUB_CATEGORY_DESC
            ⊞ WAREHOUSE
        ⊞ Macros
        ⊞ Procedures
        ⊞ Functions
    ⊞ UASW103
```

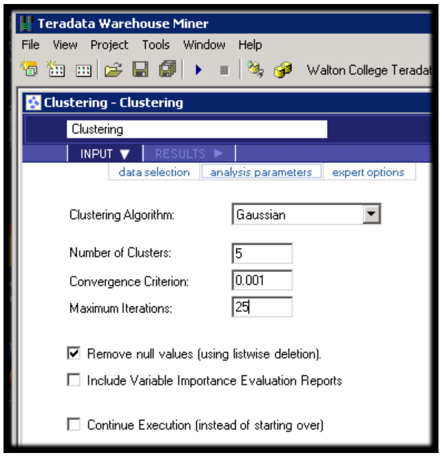
Query (Walton College Teradata)

```sql
SELECT count (*)
FROM UA_SamsClub.member_index
```

Answerset 1

	Count(*)
1	5,668,375

Figure 3. Getting a count of how many unique Sam's Club members there are (about 5.66 million)



Clustering

Clustering Algorithm: Gaussian

Number of Clusters: 5

Convergence Criterion: 0.001

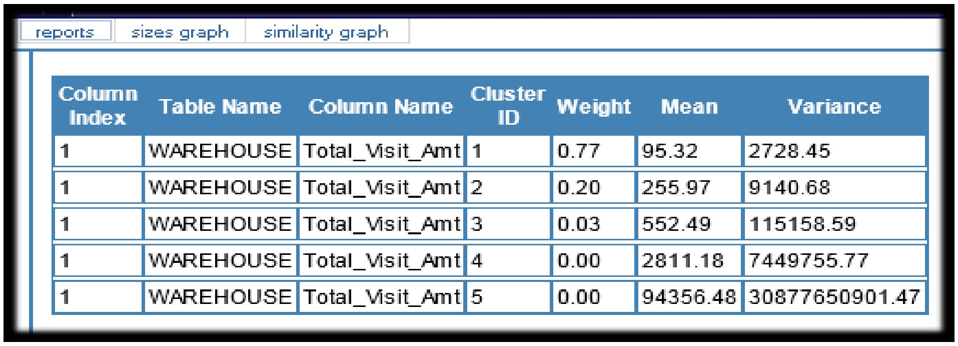
Maximum Iterations: 25

☑ Remove null values (using listwise deletion).

□ Include Variable Importance Evaluation Reports

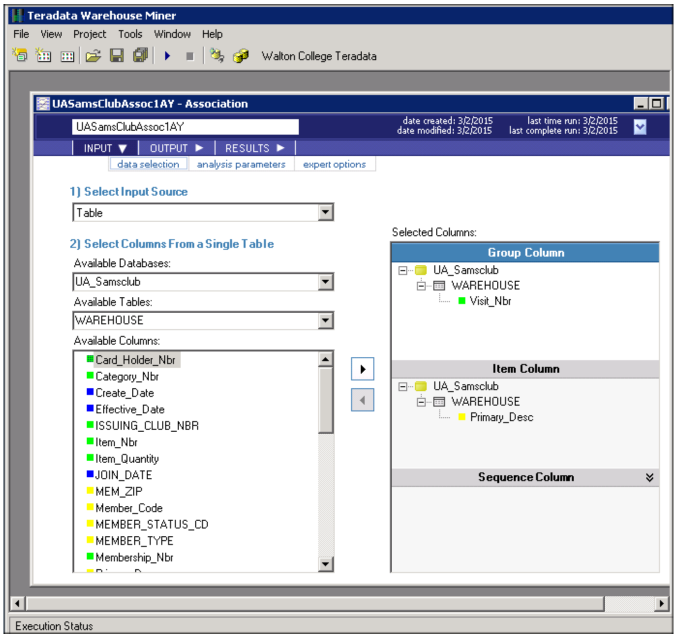
□ Continue Execution (instead of starting over)

Figure 4. Defining the number of clusters and iteration


reports
sizes graph
similarity graph

Column Index	Table Name	Column Name	Cluster ID	Weight	Mean	Variance
1	WAREHOUSE	Total_Visit_Amt	1	0.77	95.32	2728.45
1	WAREHOUSE	Total_Visit_Amt	2	0.20	255.97	9140.68
1	WAREHOUSE	Total_Visit_Amt	3	0.03	552.49	115158.59
1	WAREHOUSE	Total_Visit_Amt	4	0.00	2811.18	7449755.77
1	WAREHOUSE	Total_Visit_Amt	5	0.00	94356.48	30877650901.47

Figure 5. Table showing the cluster spending (mean) and the weight shows percentages



UASamsClubAssoc1AY

date created: 3/2/2015
date modified: 3/2/2015
last time run: 3/2/2015
last complete run: 3/2/2015

1) Select Input Source

Table

2) Select Columns From a Single Table

Available Databases: UA_Samsclub

Available Tables: WAREHOUSE

Available Columns:

- Card_Holder_Nbr
- Category_Nbr
- Create_Date
- Effective_Date
- ISSUING_CLUB_NBR
- Item_Nbr
- Item_Quantity
- JOIN_DATE
- MEM_ZIP
- Member_Code
- MEMBER_STATUS_CD
- MEMBER_TYPE
- Membership_Nbr

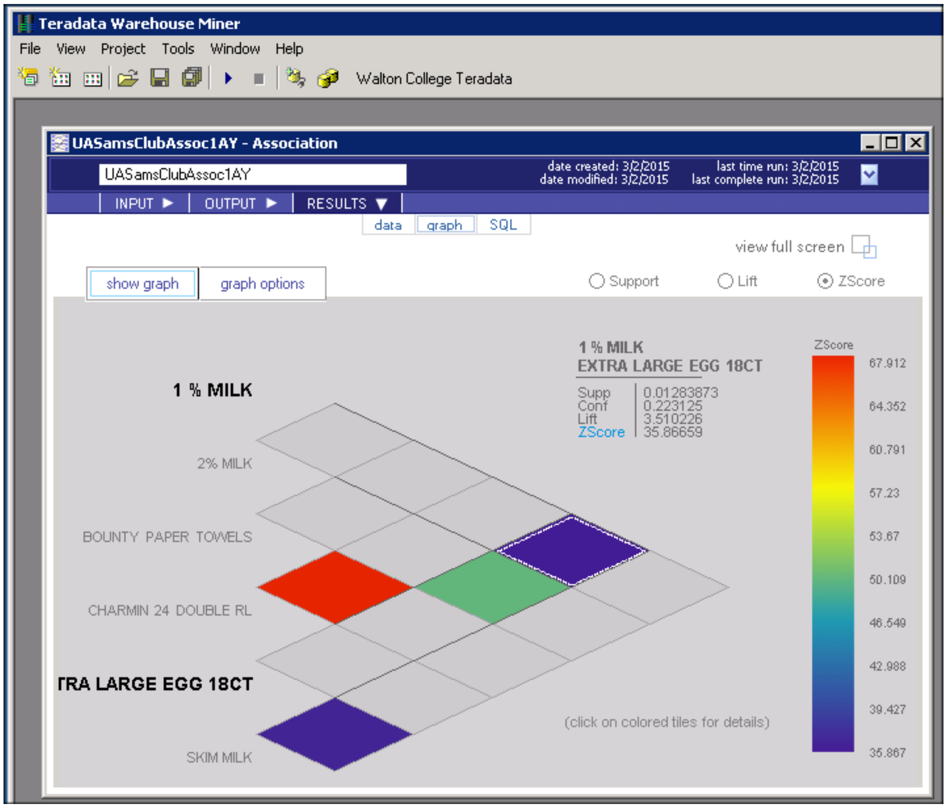
Selected Columns:

Group Column
- UA_Samsclub
  - WAREHOUSE
    - Visit_Nbr

Item Column
- UA_Samsclub
  - WAREHOUSE
    - Primary_Desc

Sequence Column

Figure 6. Choosing the Product (Primary_Desc) and the frequency people buy various products on every visit (Visit_Nbr)



UASamsClubAssoc1AY

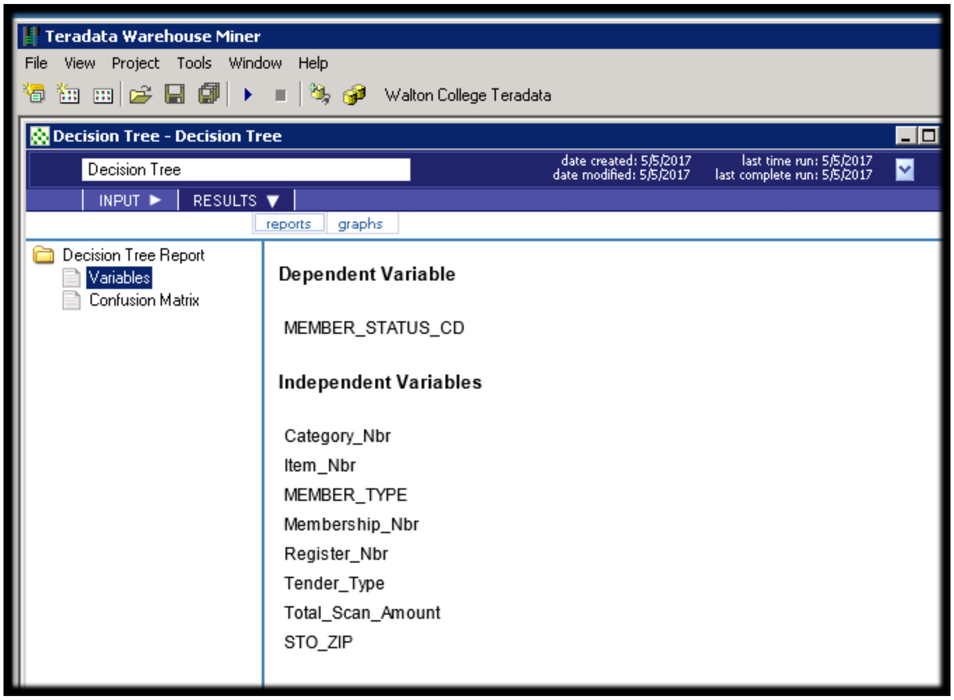
date created: 3/2/2015
date modified: 3/2/2015
last time run: 3/2/2015
last complete run: 3/2/2015

○ Support ○ Lift ● ZScore

1 % MILK
EXTRA LARGE EGG 18CT

Supp	0.01283873
Conf	0.223125
Lift	3.510226
ZScore	35.86659

Figure 7. Associative Analytics show close association between paper towels and toilet papers and eggs and milk for people who shop at this Sam's Club store.



Decision Tree

date created: 5/5/2017
date modified: 5/5/2017
last time run: 5/5/2017
last complete run: 5/5/2017

Dependent Variable

MEMBER_STATUS_CD

Independent Variables

Category_Nbr
Item_Nbr
MEMBER_TYPE
Membership_Nbr
Register_Nbr
Tender_Type
Total_Scan_Amount
STO_ZIP

Figure 8. Results shows Independent Variables that affects the Dependent Variable

"Hour of Code": A Case Study

Jie Du
dujie@gvsu.edu
Grand Valley State University
Allendale, MI, 49401, USA

Hayden Wimmer
hwimmer@georgiasouthern.edu
Georgia Southern University
Statesboro, GA, 30460, USA

Roy Rada
rada@umbc.edu
University of Maryland Baltimore County
Baltimore, MD, 21250, USA

Abstract

This study investigates the delivery of the "Hour of Code" tutorials to college students. The college students who participated in this study were surveyed about their opinion of the Hour of Code. First, the students' comments were discussed. Next, a content analysis of the offered tutorials highlights their reliance on visual programming in stylized languages with continual feedback in gaming contexts. Difficulties in delivery stem in part from the poor organization of tutorials from Code.org which makes it difficult to locate suitable tutorials. Based on the analysis of the students' comments and the content analysis of the "Hour of Code" tutorials, the authors suggest that a deeper alignment of marketing, teaching organizations, and content providers would help sustain the type of initiative exemplified by the Hour of Code.

Keywords: Advocacy, Hour of Code, code.org, online tutorials, introductory computer programming, survey, literature review.

1. INTRODUCTION

Much has been written about efforts to spread computer science education. The difficulties that students face in learning to program have been studied by Kinnunen and Simon (2012). Difficulties students encounter include required for systems development such as problem-solving, coding, and testing (Scott, 2008). A National Science Foundation sponsored study concluded that further training of computer science teachers was crucial (Goode & Margolis, 2011). The Berkeley Foundation for Opportunities in Information Technology concluded that helping underserved students appreciate computing has acquired predictable, year-round funding (Crutchfield et al., 2011). Such conclusions about historical continuity, personnel training, and funding have been re-discovered time and again as crucial to education. Many other pre-conditions for a successful education initiative could be noted.

Code.org is a non-profit organization that is dedicated to bringing computer programming into the mainstream dialogue and raising national awareness about this issue. As part of the initiative, Code.org hosted an Hour of Code curriculum of 100+, one-hour-long, computer science activities. The term "Hour of Code" may also refer to a specific, one-hour introduction to computer science that is organized by Code.org.

The "Hour of Code" activities are game-based. Students can learn computer science basics by playing a game, such as Minecraft. The "Hour of Code" tutorials teach students how to utilize problem-solving skills and logic to win the games. The "Hour of Code" tutorials are on-line, web-based, and work on computers or mobile devices. The "Hour of Code" tutorials are designed for all ages and are available in over 45 languages.

A computer science education nurtures problem-solving skills and creativity. The goal of the Hour of Code is to demystify code and show that anybody can learn the basics. Participants in the Hour of Code will hopefully learn that computer science is fun and creative. Since Code.org was launched in 2013, over 100 million students have participated in 200,000 code.org events worldwide.

How successful was the Hour of Code and what can be learned by studying what transpired? To address this question, this paper presents one study on college students to investigate how to facilitate the delivery of an Hour of Code to students. An analysis of the offered tutorials highlighted their reliance on visual programming in stylized languages with continual feedback in gaming contexts. Among other difficulties, the organization of tutorials from Code.org made it difficult to determine what tutorials to use.

The remainder of this paper is organized as follows. Section 2 reviews the published documents about the "Hour of Code" initiative. Section 3 presents a survey study on the delivery of the Hour of Code. The challenges of improving the Hour of Code are discussed in Section 4. Section 5 concludes the paper.

2. BACKGROUND

Scholarly Publications

The authors queried the ACM Digital Library, IEEE Xplore, and Google Scholar for the string "code.org" on Nov. 1, 2013. Many retrieved citations were false positives or redundancies. For instance, in the false positive category "code.org" occurred in pharmacy articles and in the redundancy category several citations were to the same interview with the founder of Code.org (Hadi Partovi). The query on the ACM Digital Library for "Code.org" on Nov. 1, 2013 retrieved 11 citations which after excluding false positives and redundancies reduced to three:

- One citation (Ardis & Henderson, 2013) referenced recent efforts to encourage coding, including quoting President Barack Obama and Code.org but then argued that such a focus on coding was counter-productive because the better effort was to teach students the principles of computer science.
- One citation was to a multi-page interview with the founder Hadi Partovi of Code.org (Snyder, 2013). Partovi emphasized his enthusiasm for the code.org vision but also that he was the only employee of Code.org and needed more time to sleep.
- Only one article (Lee, Ko, & Kwan, 2013) was a research paper but that paper only says about Code.org: "Recent press about Code.org and other efforts to increase computing literacy have begun to attract millions of people to learn computer programming."

The query on IEEE Explore returned two citations of which one was a false positive and the other was an editorial about the future of Code.org which spoke in glowing terms of the endless possibilities (Wilson, 2013).

The query on Google Scholar for "code.org" retrieved 674 citations. The first 50 citations had nothing about Code.org as related to computer programming, and most citations referred to a pharmaceutical agent. The query was modified to include the term "Partovi". Then the query retrieved exactly 6 citations. Only one of those was relevant and not already covered by the ACM and IEEE searches, and it was an online report about attracting women into computing (Mueller, 2013) and referred to Code.org as an example of an advocacy campaign for getting school-aged children interested in coding.

Library	Number of Citations		
	Retrieved	Unique and Relevant	About Advocacy
ACM Digital Library	11	3	3
IEEE Xplore	2	1	1
Google Scholar	6	1	1
Totals	**19**	**5**	**5**

Table 1. This table shows the number of citations retrieved from each library by category in 2013.

Across three retrieval engines, queries returned only a handful of relevant articles (see Table 1). More specifically, a total of 19 citations were retrieved. Five of those were unique and relevant, and all 5 were about Code.org's advocacy situation in 2013.

Another Look at Publications
The authors queried the ACM Digital Library, IEEE Xplore, and Google Scholar for the string "Code.org" again on Jan 13, 2014 and saw a doubling of the number of citations between November 2013 and January 2014 thereby demonstrating the growing interest in Code.org. A query on the ACM Digital Library for "Code.org" on Jan 13, 2014 retrieved 22 citations which after excluding false positives and redundancies included, of course, the 3 retrieved on Nov. 1, 2013 but also 6 more:

- Three Communications of the ACM editorials or business reports, one each in the Nov 2013, Dec. 2013, and Jan 2014 issues.
- Two SIGCSE Bulletin news articles in the October 2013 issue.
- An Inroads Dec 2013 opinion piece.

ACM has aligned itself with Code.org and the increasing number of publications supports the advocacy activity of Code.org. The query on IEEE Explore returned only one further citation which was an announcement that the IEEE Computer Society was a promotional sponsor of the hour-of-code (IEEE Computer, 2013). A query on Google Scholar for "Code.org Partovi" returned only two new, relevant citations: a news story on a web site and a bachelor's thesis from a Dutch university that pointed to Code.org as an example of an advocacy effort for computer education (Verkroost, 2013).

In order to dig deeper into the news story aspect, LexisNexis was accessed. A query for 'Code.org' on LexisNexis on Jan 14, 2014 returned 1,000 citations. The first 50 showed that many newspaper articles accompanied Computer Science Education Week and gave credit to Code.org. Code.org gave $10,000 grants to schools for the purchase of laptops with some grants announced during Computer Science Education Week. Extracts from a newspaper article in the Charleston West Virginia Gazette illustrated the kind of publicity generated (Charleston Gazette, 2013): "... all of the 670 students at Winfield Middle School will take part in the largest education event in history: The Hour of Code. .... To aid them in this endeavor, Code.org has awarded the school a $10,000 grant to purchase laptop computers. ... " For $10,000 to generate this much publicity is impressive. Several California newspapers noted that the Hour of Code was a publicity stunt but that a publicity stunt was needed (Cassidy, 2013): "Yes, we can all agree that this week's big Hour of Code initiative is a publicity stunt, ... A publicity stunt is exactly what we need." A few news reports showed longer-lasting, larger-scale results, such as an announcement from Chicago. Chicago announced on Dec. 10, 2013 a new initiative for Chicago Public Schools (US Official News, 2013): "In the next three years, every high school will offer a foundational 'Exploring Computer Science' course. ... Chicago Public Schools will receive free computer science curriculum and ongoing professional development and stipends for teachers to implement this plan thanks to a district partnership with Code.org, a nonprofit dedicated to increasing access to a computer science education."

The first 50 results from LexisNexis showed this pattern:

1. Eight news stories explored issues addressed by Code.org's hour-of-code,
2. Five news stories were about local schools participating in the hour-of-code (without mention of a $10,000 grant).
3. Four news articles (about different school districts in different states) noted a $10,000 grant from Code.org and the hour-of-code initiative.
4. Four news stories were from outside the US, such as one from Sri Lanka about a software company that helped several children experience a Code.org tutorial (Daily Mirror, 2013).
5. Three were official announcements, one from the City of Chicago, one from Microsoft, and one from the Patent and Trademark Office (which noted that Code.org trademarked 'CODE').

The remaining 26 citations were either irrelevant or redundant.

Publications in More Recent Years
The authors queried the ACM Digital Library for the string "Code.org" again on June 6, 2017. The query on the ACM Digital Library for "Code.org" retrieved 45 citations which after excluding false positives and redundancies from previous searches reduced to 13:

- One Communications of the ACM editorial in the Feb. 2017 issue.

- One SIGCSE Bulletin news article in the January 2015 issue.
- Seven ACM Inroads opinion pieces from Dec. 2014 to Nov. 2016.
- Two abstracts from ACM technical symposium on Computer science education proceedings. Hanley (2016) discussed using Myna, a Programming by Voice tool to support children with a mobility disability to experience Code.org's educational opportunities. Meeker (2014) dsicussed the need to better promoted the "Hour of Code" initiative in local schools.
- Two research papers discussed how the hour of code can be utilized to improve the learning process. Piech, et al. (2015) developed a family of algorithms that can predict the way an expert teacher would encourage a student to make forward progress and then used the algorithms to automatically generate hints for the Hour of Code. Theodoropoulos et al. (2016) assessed the learning effectiveness and motivational appeal of the Code.org's activity named K-8 Intro to Computer Science. Seventy-seven students of two Greek high schools participated their study and the results show that these specific educational computer games provide a high-quality learning experience.

Across these document databases, one sees that Code.org received substantial publicity for its "Hour of Code" initiative. News stories corroborate the initiatives of Code.org which are to train teachers and to encourage states to consider computer science education mainstream.

3. UNIVERSITY CLASSROOM SURVEY

A study was conducted to investigate how educators could deliver the Hour of Code to students. The purpose of this study is to 1) examine the effect of the Hour of Code on students' attitude toward programming, and 2) gain insights on improving the "Hour of Code" initiatives to better promote computer science education.

Lesson Plan

One of the authors has been teaching introductory computing courses at one Midwestern public, master's granting university for years. Since summer 2014, the author started introducing the "Hour of Code" tutorials in the introductory computing course. The author usually started the class with a 6-min video that begins with this quote from Steve Jobs "Everybody in this country should learn how to program a computer because it teaches you how to think" and follows with multiple short interviews where people discussed the role of computers in their life and how important they believed it would be that everyone learn to code for, at least, an hour. The interviewees represented a broad spectrum of types of people well-known to Americans and included Bill Gates, the founder of Microsoft, and Chris Bosh, a famous American basketball player. Next, students were asked to undertake the tutorial "Write Your First Computer Program" from the category of "Tutorial for Beginners" at code.org. Students were asked to take a pre- and post-survey about their experience at Code.org.

Data Collection

This practice was first implemented in 2014 and then has been repeated every year since then. Thus, data was collected from 2014 to 2017 from the students enrolled in an introductory computing course at one Midwestern public, master's granting university. As a result, the data set yielded 255 usable responses (116 in 2014, 45 in 2015, 47 in 2016, and 47 in 2017). Most of the participants are freshmen and fresh-women with a spectrum of majors including business, accounting, criminal justice, allied health sciences, geography, hospitality tourism management, and psychology.

Both qualitative and quantitative data were collected. The quantitative data were collected using Likert-Scale survey questions. The quantitative data analysis from the 2014 data set shows that the Hour of Code has a positive impact on students' attitude toward programming. The detailed survey questions and results from the 2014 data set can be referenced at (Du, Wimmer, & Rada, 2016).

For qualitative data collection and the focus of this paper, participants were asked to provide additional comments regarding programming as well as the "Hour of Code" tutorials at the end of each survey. As a result, 39% of the participants provided additional comments. That is, 97 qualitative comments (55 from the pre-survey and 42 from the post-survey) were collected.

Data analysis

The authors analyzed the students' feedback on their experience at Code.org. Students' feedback on their attitude toward programming was first presented. A discussion on challenges of delivering the hour of code follows.

Attitude toward Programming

Students' comments in the pre-survey before they completed the "Hour of Code" tutorial are incredibly negative (see Figure 1). The top 3 high-frequency quotes show the participants' frustration:

- *I have no clue what I am doing. (25%)*
- *No previous programming experience. (23%)*
- *I am confused. (16%)*

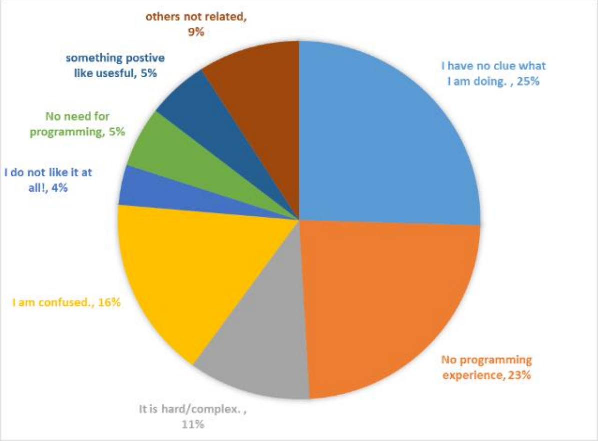


Figure 1 Participants' comments in the pre-survey

It is interesting to find that participants showed more interests on programming after completing the "Hour of Code" tutorial at Code.org (see Figure 2). Although some comments address the difficulty of learning to program, most of the comments (nearly 70%) are very positive. The selected quotes from students' comments in the post-survey illustrate that students appreciated the "Hour of Code" tutorial:

- *I have no realized programming is not just a bunch of symbols on a computer, it is a way of thinking, processing, and problem solving*
- *I think it is interesting.*
- *That was a very cool and helpful assignment. I have never done anything like that before.*
- *I am interested in coding quite a bit, it can unlock a lot of potential!*
- *Programming seems a lot easier.*
- *The tutorial put it into terms that were easier to understand.*
- *This seemed to simplify programming for me.*
- *I may look into programming more.*
- *Learned a bit more about coding.*
- *Interesting tutorial, enjoyed programming.*
- *Programming seems very helpful for many occupations.*
- *I have been playing Minecraft for many years but not like this! Very cool!*

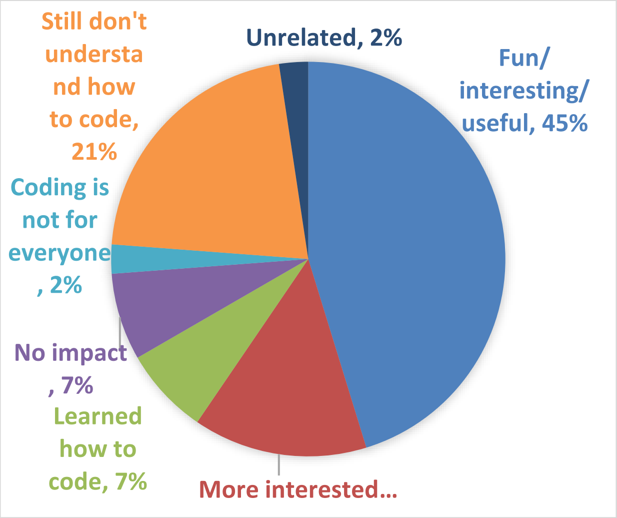


Figure 2 Participants' comments in the post-survey

Observations of Delivering the Hour of Code

One challenge the authors faced has been accommodating the widely different skill levels of students in the class. Some students had very limited knowledge about coding, while other students already took a computing course in high schools. The authors appreciate the difficulty to identify an appropriate "Hour of Code" tutorial for the college students with widely different coding skills. The selected quotes coming from two ends of this spectrum highlight this challenge:

- *It is cool and fairly easy.*
- *I already am familiar with coding languages, not much impact.*
- *It is complicated.*
- *Very interesting, but I still feel like I don't understand very much about programming code.*

Another observation from the authors is that compared to their male peers, female students are more intimated by coding and reluctant to try it at the beginning. This might be supported by the fact that the gender gap in computing is getting worse nowadays. If nothing changes the way that we teach computing to girls, the gender gap is expected to hurt U.S. economy seriously. Schools and colleges need to initiate outreach efforts to have more women engaging with computer science at the same rate as men. One student's feedback echoed this effort: "*I think it is great to encourage girls to become interested in coding.*" Gender differences will be analyzed and presented in future work.

The authors also identified several things that Code.org might improve in the future. While focused on K-12 education, Code.org claims that their curriculum and tutorial materials can be used for any non-commercial, computer science educational purposes, such as introductory course in colleges. There are no appropriate tutorials designed for college students as of yet on Code.org. Some participants' feedback pointed out the lack of tutorials for college students.

- *I did relatively enjoy code.org, yet it was a little childish.*
- *It is a good tutorial for children in the aspect of coding. For young adults and college students, I think it would be more beneficial if they learned a starter language like Java.*

4. CHALLENGES OF IMPROVING THE HOUR OF CODE

In this section, a content analysis on the "Hour of Code" tutorials is presented and challenges on improving the "Hour of Code" initiatives are discussed.

Organization of Tutorials

The authors analyzed the content evolution on code.org since 2014. On Jan. 13, 2014, the top level of the organization of tutorials was 7 revolving menus that categorize tutorials as:

1. for beginners
2. learn JavaScript
3. no device
4. apps for a device
5. other programming languages
6. make your own apps
7. other learning options

Within each category, each tutorial was presented with a handful of attributes to include most prominently the author, the title, a paragraph blurb, the target ages, the number of participants, and the URL. Code.org authored the tutorial called "write your first computer program" which is the famous "Angry Birds" tutorial and which had the most participants at 9,900,000.

The catalog listed 30 tutorials, although some tutorials occurred in two categories (see Appendix). For instance, MIT's "App Inventor" was listed under both the "for beginners" and the "make your own apps" categories. Tynker's "build your own game" was listed under "for beginners" but also under "other learning options". In taxonomy construction, one should define each tutorial and category by attribute values and enter a tutorial under a category when it inherited attribute values from its parent. Such a taxonomy policy was not adequately followed in the design of menus for these tutorials.

The number and the organization of the "Hour of Code" tutorials have dramatically changed since 2014. On June 6, 2017, the catalog listed 149 tutorials which can be further filtered by:

- Grades
  - Pre-readers
  - Grades 2-5
  - Grades 6-8
  - Grades 9+
- Educator experience
  - Beginner
  - Comfortable
- Student experience
  - Beginner
  - Comfortable
- Classroom technology
  - Computers
  - Android
  - iPad/iPhone
  - Poor or no internet
  - No computers or devices
- Topics
  - Science
  - Math
  - Social Studies
  - Language Arts
  - Art, Media, and Music
  - Computer Science only
- Activity type
  - Self-led tutorials
  - Lesson plan
- Length
  - One hour
  - One hour with follow on
  - A few hours
- Language
  - Blocks
  - Typing

Although the new organization incorporates more factors to facilitate locating a tutorial, a teacher might still struggle to identify which tutorials would be appropriate for which students. For instance, it is unclear how the beginner and comfortable levels were defined. Some educators might have a hard time locating appropriate tutorials due to lack of understanding about the organization and how it is defined.

Challenges

The Code.org organizers claimed that the Hour of Code was enormously successful. An email sent on Dec. 23, 2013 from Hadi Partovi (the creator of Code.org) to his supporters began with "Thank you for supporting computer science and making the Hour of Code an amazing success. In 2 weeks, 20 million students worldwide learned an Hour of Code, including 1 in 4 US students, and half were girls!" However, neither the email nor the web site explains how the number of participants was determined. One does not need to log into a tutorial in order to use it. The authors visited the "Angry Birds" tutorial many times over different days from different computers and different countries. The server would have considered those visits to come from different participants but that overestimates the number of unique participants. The Dec. 23rd email concludes by encouraging readers to ask Congress to support the Computer Science Education Act, to ask their local schools to teach computer science, and to donate money or time to Code.org.

Many citizens of the world see the benefits that could accrue from a wider dissemination of computer knowledge. Code.org's goals would ultimately be best achieved by the government, as Code.org admits. Computer science education reaches a larger fraction of the population in many countries than it does in the US. For instance, the Kingdom of Saudi Arabia has nationalized plans that put computer labs in every secondary school and trained 6,000 teachers of computer science (Ministry of Education, 2013). In Singapore, the Ministry of Education has a country-wide, detailed master that provides a student-to-computer ratio of 2:1 in all public schools and a curriculum that is extensively supported by information technology and teaches information technology (Committee on Compulsory Education, 2000). The management of K-12 education in the US does not seem to support such centralized, country-wide results.

Implications and Limitations

This study investigates the delivery of the Hour of Code tutorials to college students. Students are not expected to become an expert of computer science in one hour. The Hour of Code is only the first step for students to learn that computer science is fun and creative. The findings of this study show that the participants became much more interested in programming after they tried the tutorials and expressed an interest to know more about coding. Educators can gain confidence after learning the "Hour of Code" tutorials that they can teach computer science even though they may not have a computer science degree. It is important to learn that everyone can learn computer science.

Nowadays computers are used in almost every industry. According to National Center for Education Statistics, there were about 60,000 computer science graduates from US institutions in 2015 and there were about 530,000 computing job opening at that year. It is estimated that by 2020 there will be 1.4 million computing jobs available while the number of computer science graduates will only be 400,000. The gender gap makes things worse. According to Accenture and Girls Who Code, 24 percent of computing jobs are held by women and this number will fall to 22% by 2025 if no changes are made on how computer science is taught to girls. It is vital for educators to 1) help students understand the importance of computer science and technology to their lives and their future career paths, and 2) teach students computer science in an easy to understand way to help them gain problem solving skills as an essential skill for life. Educators could explore the various "Hour of Code" activities and may find ones that fit in their curriculum and meet their students' need.

There are some limitations worth noting with this study. This study was limited to students at a single university. The number of survey questions are limited. Given that, various interpretations on the findings could be argued. A more diverse sample will increase the validity. More questions in the survey will help better understand participants' attitudes toward programming and their computer science skills. Gender differences on learning coding will be analyzed in future work.

5. CONCLUSIONS

The Hour of Code is claimed by its organizers to be a success. A review of the literature shows that much of what has been written about Code.org identifies it as an advocacy organization. A survey study of college students learning the Hour of Code shows that the Hour of Code stimulated students' curiosity and opened their mind to programming.

The initiative to provide an Hour of Code could be improved, and one improvement would be to better coordinate the advocacy and the tutorial delivery activities. The authors attempted to work with local colleges to institute an "Hour of Code". However, the instructors needed to reduce the scope of their plans, in part because of the difficulty of identifying the appropriate tutorial to use with their students. A better catalogue of tutorials and semi-automated aids to match a student to the appropriate tutorials might help.

While, as with any qualitative study, different researchers may come to different conclusions, we aim to illustrate our experiences and entice other instructors to also consider Hour of Code in their classrooms. Educators can host an Hour of Code all year round. A good time to host an Hour of Code is the annual Computer Science Education Week (such as the one December 4-10, 2017).

6. REFERENCES

Ardis, M. A., & Henderson, P. B. (2013). Software Engineering != Coding. *ACM SIGSOFT Software Engineering Notes, 38*(3), 5.

Cassidy, M. (2013, December 12). Code.org Hour of Code builds a deeper understanding of the power of computing, *Contra Costa Times*.

Charleston Gazette. (2013, December 18). Winfield Middle obtains grant to teach coding, *Charleston Gazette (West Virginia)*.

Committee on Compulsory Education. (2000). *Report of the Committee on Compulsory Education in Singapore*. Singapore: Ministry of Education. Retrieved from http://www.moe.gov.sg/initiatives/compulsory-education/files/ce-report.pdf.

Crutchfield, O. S. L., Harrison, C. D., Haas, G., Garcia, D. D., Humphreys, S. M., Lewis, C. M., & Khooshabeh, P. (2011). Berkeley Foundation for Opportunities in Information Technology: A Decade of Broadening Participation. *ACM Transactions on Computing Education, 11*(3), 1-24.

Daily Mirror. (2013, December 24, 2013). Brandix i3 supports global software coding initiative, *Daily Mirror of Sri Lanka*.

Du, J., Wimmer, H., & Rada, R. (2016). "Hour of Code": Can it change students' attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice, 15*, 52-73.

Goode, J., & Margolis, J. (2011). Exploring Computer Science: A Case Study of School Reform. *ACM Transactions on Computing Education, 11*(2), 1-16.

Hanley, C. R. (2016). *Programming by Voice to Support Hour of Code for Children with Motor Disabilities (Abstract Only)*. Paper presented at the Proceedings of the 47th ACM Technical Symposium on Computing Science Education, Memphis, Tennessee, USA.

IEEE Computer. (2013). Hour of Code Kicks Off to Introduce K-12 Students to Computer Programming. *Computer, 46*(11), 99.

Kinnunen, P., & Simon, B. (2012). My program is ok – am I? Computing freshmen's experiences of doing programming assignments. *Computer Science Education, 22*(1), 1-28.

Lee, M. J., Ko, A. J., & Kwan, I. (2013, August 12-14, 2013). *In-Game Assessments Increase Novice Programmers' Engagement and Level Completion Speed.* Paper presented at the ICER'13 Ninth Annual International ACM Conference on International Computing Education Research San Diego, California.

Meeker, P. H. (2014). *Inspiring a love of computer science through the education of our youth (abstract only)*. Paper presented at the Proceedings of the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA.

Ministry of Education. (2013). IT Department Retrieved January 14, 2014, 2014, from http://www2.moe.gov.sa/english/Pages/it_department.htm

Mueller, R. (2013, March 21-22, 2013). *Attracting Females into ICT in Canada* Paper presented at the Information and Communications Technology (ICT) Talent Workshop, Ottawa, Canada.

Piech, C., Sahami, M., Huang, J., & Guibas, L. (2015). *Autonomously Generating Hints by Inferring Problem Solving Policies*. Paper presented at the Proceedings of the Second (2015) ACM Conference on Learning @ Scale, Vancouver, BC, Canada.

Scott, E. (2008). From Requirements to Code: Issues and Learning in IS Students' Systems Development Projects. *Journal of Information Technology Education: Innovations in Practice, 7*, 1-13.

Snyder, L. (2013). An Interview with Hadi Partovi. *Communications of the ACM, 56*(9), 41-45.

Theodoropoulos, A., Antoniou, A., & Lepouras, G. (2016). How Do Different Cognitive Styles Affect Learning Programming? Insights from a Game-Based Approach in Greek Schools. *Trans. Comput. Educ., 17*(1), 1-25. doi: 10.1145/2940330

US Official News. (2013, December 10). Illinois: Mayor Emanuel And CPS CEO Barbara Byrd-Bennett Announce Comprehensive K-12 Computer Science Program For CPS Students, *US Official News*. Retrieved from http://www.cityofchicago.org/

Verkroost, Y. (2013). *Seriousify and prettify our educational system! ABOUT THE USE OF (SERIOUS) GAMES AND AR IN EDUCATION*. Bachelor Project VU University Amsterdam. http://www.yordiverkroost.nl/vu/paper.pdf, Amsterdam.

Wilson, C. (2013). What's up next for Code.org. *IEEE Computer, 46*(8), 95-97.

Appendix

This appendix shows the complete taxonomy of the tutorials at http:code.org/learn retrieved by the authors on Jan. 13, 2014. The 7 left-most list items are the categories. The numbered sub-lists give the author of the tutorial followed in quotes by the title of the tutorial.

For Beginners

1. code.org "write your first computer program"
2. Scratch "create a holiday card"
3. tynker "build your own game"
4. lightbot "lightbot"
5. MIT Center for Mobile Learning "AppInventor Hour of Code"

JavaScript

1. CodeCombat "CodeCombat"
2. KhanAcademy "Introduction to Javascript"
3. codeacademy "codeacademy"
4. codeHS "Learn to Code with Karel the Dog"
5. code avengers "Learn to Code a javascript quiz game"

No Device

1. Thinkersmith "My Robotic Friends"
2. Google Education "Blockly"
3. Thinkersmith "Binary Baubles"
4. Kodable "fuzzFamily Frenzy"

Tutorial Apps

1. Lightbot "Lightbot"
2. Kodable "Kodable"
3. Hopscotch "Code on your Ipad"

Other Programming Languages

1. Grok Learning "A Taste of Python Programming"
2. Processing "Drawing with Code"
3. Robomind Academy "Program a virtual robot"
4. MakeGameswithus "Build an iPhone game in your browser!"

Make Your Own Apps

1. MakesGameswithus "Build an iPhone game in your browser"
2. MIT Center for Mobile Learning "AppInventor Hour of Code"
3. Microsoft Research "TouchDevelop"

Other Learning Options

1. Tynker "Build your own Game"
2. Microsoft Research "Touch Develop"
3. University of Colorado "Make a 3D Frogger Game in an hour"
4. Alice Project "Intro to Programming with Alice 2"
5. RunRev "Everyone can code with LiveCode"
6. Washington University Computer Science "Looking Glass"

ViNEL: A Virtual Networking Lab for Cyber Defense Education

Bryan Reinicke
breinicke@saunders.rit.edu
Rochester Institute of Technology
Rochester NY 14623

Elizabeth Baker
bakere@uncw.edu
University of North Carolina Wilmington
Wilmington, NC 28403

Callie Toothman
callie.toothman@GE.com
General Electric Digital Technology

Abstract

Professors teaching cyber security classes often face challenges when developing workshops for their students: How does one quickly and efficiently configure and deploy an operating system for a temporary learning/testing environment? Faculty teaching these classes spend countless hours installing, configuring and deploying multiple system configurations only to decommission the operating system after needed usage is complete. This paper presents a system that was developed to significantly reduce testing environment setup times by establishing a virtual networking lab (ViNeL) that allowed a cyber defense club at a public regional university to spin up an entire network of virtual machines to simulate business environments and prepare for competitions. The application also allowed club members to launch a single operating system for lab style experimentation. The application uses the remote desktop protocol (RDP) to make it platform independent. Built using Oracle VirtualBox on a LAMP stack (Linux, Apache, MySQL, and PHP), the virtual networking lab is completely open source and can be implemented on a variety of server hardware configurations. Details on the development and implementation are presented to aid in the development of similar systems in different environments.

Keywords: Cyber Security, virtual environments, curriculum, building systems.

1. INTRODUCTION

Professors teaching cyber security classes often face challenges when developing classes for students: How does one quickly configure and deploy an operating system for a temporary testing environment and store the configuration for later use? As is the case with most enterprise level system administrators, most instructors spend countless hours installing, configuring and deploying multiple system configurations. However, unlike system administrators, the configurations are used for lab exercises and practice sessions. Then they destroy the lab OS configurations after the needed usage is complete to make room for the next lab activity and subsequent setup.

ViNeL was created to provide a Virtual Networking Lab to allow cyber defense club members and team competitors' constant availability to operating systems and configurations used in workshops and practices without the need to rebuild OS configurations from scratch. Using a private cloud infrastructure, the application also decreased the configuration deployment time from an average of 5 hours to 30 minutes; a reduction of 90%. This same technology could be applied to classroom environments in different areas.

This paper will explain the setup used by the cyber defense club at a public regional university, highlighting the challenges the club faces with lab configuration. This is followed by a review of possible technologies that can be used to create such a proposed networking lab. After detailing the implementation solution, the results of the implementation are reviewed. Lastly, we discuss the implementation of such structures for testing and development.

2. PROBLEM CONTEXT

In this section we outline the typical setup process used by the cyber defense club and highlight any requirements for the new system in order to address the challenges faced in the lab and operating system setup process.

CYBER DEFENSE CLUB Setup Processes

Since the CYBER DEFENSE CLUB is comprised of a club members and a competition team, each segment of the CYBER DEFENSE CLUB has specific needs for system setups.

Club: Lab Use

The CYBER DEFENSE CLUB holds workshops every 1-2 weeks. These workshops use multiple configurations across various operating systems that get used from year-to-year. However, there is currently no organization in the archival of such configurations. Thus, lab configurations are implemented from scratch for each use. The following setup process is used by club officers to setup computers for workshops and labs.

1. Installation – The operating system is downloaded or retrieved from a network storage device and installed on the master machine. This master can either be a virtual machine or a physical computer. Each operating system is installed from an online repository which uses a minimal copy of the operating system. The time of installation was increased as most of the installation files will have to be downloaded at the time of installation. However, all current versions of packages are installed and no updates will need be applied during the next phase of setup. This method is advantageous for workshop configurations that need updated software for practicum. (Timeframe: 15 – 60 minutes)
2. Configuration – Each system is configured for the needed services required by competition practice or workshops. New applications are installed and tested to ensure execution of lab steps. (Timeframe: 1 – 5 hours)

3. Distribution – Once the master machine has been properly configured, the image is distributed using one of the following methods. (Timeframe: 1 – 8 hours)
a. VirtualBox + Dropbox – If the installation was created on a virtual machine, the entire VM is exported in the Open Virtualization Format (OVF) via Oracle VirtualBox (Oracle VM Virtualbox, 2015) and added to the CYBER DEFENSE CLUB Dropbox. This method was preferred as the local installation of Dropbox on each member machine automatically downloads an offline copy of the VM that can later be imported into VirtualBox. However, Dropbox has a storage limit and as such this method is typically used for smaller VMs.
b. Clonezilla – If the installation master is created on a physical machine, an image file is created using the disk imaging tool Clonezilla and stored to the CYBER DEFENSE CLUB network attached storage device. The master is then physically cloned to each machine using the Clonezilla interface. This method is used for VMs that are too large to upload to Dropbox. However, it is not preferred as all previous configurations on the client machine are erased.

A summary of the lab use setup is shown in **Figure 1** (Contained in the appendix).

Team: Competition Practice Use

The CYBER DEFENSE CLUB competition team competes in the Southeast Collegiate Cyber Defense Competition (SECCDC). Each year, the SECCDC releases two network topologies used in competition (SECCDC, 2015). Competition topologies contain 5 – 8 networked machines each. The CYBER DEFENSE CLUB creates an in-house copy of these topologies on which to practice throughout the competition season. These systems were created on physical machines and erased at the end of each season. The following setup process for each machine is used by competition team members for competition practice.

1. Installation – The operating system is downloaded or retrieved from a network storage device and installed from a full installation disc which uses full copy of an operating system saved to disk as the source of installation. Due to teams being responsible for updating and patching their

systems during competition, this method is more convenient for competition network setups. (Timeframe: 15 – 60 minutes per machine)

2. Update – Once properly configured to access the internet, the operating system is patched and updated to the latest available versions by the team member responsible for that OS. (Timeframe: 1 – 5 hours per machine)

3. Configuration – Once updates have been applied, each system is configured for the needed services need for competition practice or workshops. (Timeframe: 1 – 5 hours per machine)

4. Networking – Once the network domain is properly configured, the networking configuration is applied to each system. (Timeframe: 5 – 30 minutes per machine)

A summary of the competition practice setup is shown in **Figure 2** (in the Appendix).

Requirements

The purpose of this project is to find a solution or build an application to ease the lab setup process for the CYBER DEFENSE CLUB and its team. Due to the differences in lab and practice setups, the OS deployment application system will need to be developed into two major components: single instance use for club workshops and network use for competition practice. Each of the requirements outlined below will describe the necessary capabilities for each situation.

1. Performance – Each user will need to access his/her own virtual machine that meets the minimum system requirements outlined for the operating system in use.
a. Club workshops: The CYBER DEFENSE CLUB has an average attendance of 15 members per week. The application must be able to support up to enough machines to allow each attending member to run a VM simultaneously. Lab-based setups do not currently require more than 1 CPU core. (Minimum: 30 GB of RAM, 1 CPU Core)
b. Competition Practice: Competition topologies use an average of 7 servers per competition. The application must be able to support the running of all VMs configured in the network topology simultaneously. The requirements assume that only one topology is deployed at a time. Some virtual machine require the use of multiple cores. (Minimum: 20 GB of RAM, 2 CPU Cores)
The hardware used for the application will need to have a minimum of 2 cores and 30 GB of RAM.

2. Self-Service – The application will need to allow users to quickly deploy and control their own virtual machines and networks without the need for administrative assistance or manual installation. Each user will need their own account to ensure access control for each running VM instance.

3. Platform Independent – Members of the CYBER DEFENSE CLUB use Windows, Mac and Linux systems for personal use. Allowing users to access VMs from personal laptops will reduce the amount of hardware needed to serve as client machines for the club. The application must be able to be run from multiple platforms, most notably Windows, Mac, and Linux operating systems.

4. Networking – For the competition team, practice servers are configured in a network provided by the SECCDC. Virtual machines within the application will need to have the ability to be networked in small, isolated, and non-conflicting networks for competition practice use.

5. Affordable – The CYBER DEFENSE CLUB has a limited budget. To avoid paying for licenses and additional hardware, the implementation cost of the application needs to be minimal. If possible, software used should have free or open source licenses.

3. CLOUD COMPUTING PLATFORMS

This chapter will review possible software solutions based on the requirements for the CYBER DEFENSE CLUB. Each product was reviewed and analyzed to determine the most suitable solution for the storage of system configurations and fast, on-demand access of those stored configurations.

Infrastructure as a Service

According to the National Institute of Standards and Technology (NIST), Infrastructure as a Service is defined as:
“The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications.” (Mell & Grance, 2011)
Of the service models listed in the NIST publication defining cloud computing, the Infrastructure as a Service (IaaS) model aligned best with the needs of the club. IaaS encapsulates the self-service requirement for the CYBER DEFENSE CLUB virtual networking lab application

and will serve as the model of service provided to the club.

Infrastructure as a Service Provider Reviews

Three IaaS providers are reviewed below. A brief overview of each is given, as well as a synopsis of how they can be used as a solution for virtual networking with the CYBER DEFENSE CLUB.

Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud (EC2) is web-based service that allows users to create virtual hosts in the cloud (Amazon EC2, 2014). Features of EC2 include:

- Pre-configured templates for deployment called Amazon Machine Images (AMI)
- Configurable network and security settings
- Management tools for VM control
- Ability to run instances in multiple locations called Availability Zones
- Pay-per-hour billing

Amazon EC2 has the ability to allow each CYBER DEFENSE CLUB user to create and manage their own virtual servers, save configurations, and re-deploy as needed. The competition team can also manage their own machines, while configuring the networking and security settings to mimic those of the SECCDC network topologies.

Microsoft Azure Virtual Machines

Microsoft Azure Virtual Machines (Virtual Machines, 2015) lets users deploy various computer configurations instantly. Features of Microsoft Azure Virtual Machines include:

- Scaling up to 1000 instances
- Virtual networking and load balancing
- Ability to transfer machines between Hyper-V and Azure
- Pay-per-minute pricing

Microsoft Azure Virtual Machines could allow the CYBER DEFENSE CLUB to create in-house images that can be uploaded to Azure for instant distribution via the Azure Virtual Network. The Azure Virtual Network can also be used to create copies of the SECCDC topologies for practice sessions.

Apache Virtual Computer Lab

Apache Virtual Computer Lab (VCL) is an open-source cloud computing platform that allows users to deploy custom computing environments (The Apache Software Foundation, 2015). Features of Apache VCL are listed below:

- Free and open source
- Ability to schedule the usage of VM instances
- Image revision control
- Provides user access control models
- Supports the use of multiple provisioning methods

Apache VCL can allow the CYBER DEFENSE CLUB to launch images based on member roles. Team members could have separate access to particular VMs to which regular club members to do not have access.

Infrastructure as a Service Provider Analysis

Each service provider was rated on how well they could meet the requirements needed for a virtual networking lab. The summary of their ratings are shown in Figure 3 (contained in the appendix).

Performance

Amazon EC2 and Microsoft Azure VMs both provided cloud-based virtual machines that could provide performance at nearly any level; however, the better the performance of the virtual machine, the higher the cost. The Apache VCL platform must be installed on your own hardware, so the performance depends upon the available resources of the user. VCL is limited to hardware capabilities on-hand, whereas EC2 and Azure VMs can be scaled elastically to meet dynamic needs.

Self-Service

Each provider allowed the user to log into an account from which they could launch and control virtual machines in a self-sufficient manner. Apache VCL trailed slightly behind due to the fact that it does not offer nearly as many features as EC2 or Azure VMs. However, not all of the additional features provided by those two IaaS platforms are necessary to meet the requirements of the CYBER DEFENSE CLUB.

Platform Independent

All platforms provided access to the virtual machines via Secure Shell protocol (SSH) or Remote Desktop Protocol (RDP), which allows access from Windows, Linux or Mac. These protocols also allow for mobile to be used for connection provided the mobile device application store includes SSH and RDP mobile applications.

Networking

Microsoft Azure VMs provided the most configurable and user-friendly networking design of the three platforms. Apache VCL did not provide a networking feature capable for use by the competition team. While there was a clustering feature which could be used for lab configurations, there was no way to network two VM instances running different operating systems in Apache VCL.

Affordable
While Apache VCL is free and open source, the Amazon and Microsoft pricing model calculated charges for per hour that the virtual machine was running. Costs were also accumulated for the storage of virtual machine images. Table 1 shows the cost of using cloud-based virtual machines for both the club member and team competitors (shown in the appendix). It is assumed that virtual machines are powered off when not in use.

Solution Selection
Based on the rating scale, Microsoft Azure VMs scored best on average for the CYBER DEFENSE CLUB requirements. Apache VCL could not be used due to the lack of networking capability. Amazon Web Services only provided a core set of servers at the above prices. Customized or unconventional operating systems incurred an additional cost.

While the cost of using Microsoft Azure VMs is reasonable, it was noted in the pricing plan that operating systems licensing fees were included in the cost. Through the Microsoft Imagine Subscription (previously Microsoft DreamSpark) purchased by the university, the CYBER DEFENSE CLUB already has free licenses to several Microsoft operating systems and software, as long as the software is installed on servers and computers located on campus (Microsoft DreamSpark Premium Subscription Agreement, 2015). This is common on campuses, as Microsoft offers this service to campuses at a very reasonable cost.

Instead of paying for licenses already freely available to students, it is feasible to create a custom platform similar to Amazon EC2, Microsoft Azure VMs and Apache VCL that meet all needs of the CYBER DEFENSE CLUB without any additional purchases.

4. IMPLEMENTATION

The custom virtual networking lab platform (ViNeL) was designed to meet all of the requirements of the CYBER DEFENSE CLUB. The system was installed using a LAMP stack (Linux, Apache, MySQL, PHP) and will allow club officers to upload configured VMs to the server to be stored and made available to use. Club members were able to launch up to one VM instance at a time. Competition team trainers can upload configured VMs used in SECCDC topologies and provide IP address details for the system to use to network the machines together. The method of implementation is detailed below. The ViNeL architectural diagram is shown in Appendix A as figure 4.

Hardware Specifications
ViNeL was installed on a Sun Microsystems Sun Fire X4150 Blade Server. The system specifications are listed in table 2 (in the appendix).

This particular server was used as it was currently available for use for the CYBER DEFENSE CLUB. However, the application can be installed on any server hardware with at least 4 CPU cores and 15 GB of RAM.

Operating System
Ubuntu Server 14.04 LTS (64-Bit) was installed on a 30 GB root partition. The rest of the hard drive space was configured to support a logical volume manager (LVM). The installation used a network installation disc with the Basic Ubuntu Server, LAMP Server, OpenSSH Server packages selected to be installed initially. The server was configured to install security updates automatically.

Web Server
The most current release of Apache HTTP server (as of writing: 2.2.29) was installed on the server. The website design was based on the Twitter Bootstrap Dashboard template (Bootstrap Core Team, 2015) and used Chart.js (Downie, 2015) to display charts of system information. Plupload (Moxiecode Systems AB, 2015) was used to upload VM images to the server by chunking the files to allow uploads to be paused and resumed. The webserver uses PHP and JQuery for website scripting. The HTTP servers were hardened using the "Security hardening on Ubuntu Server 14.04" guide (Brock, 2014).

Database
The Ubuntu MySQL server package (as of writing: 5.5) was installed on the server blade. A database was created to store user, group, and VM data. The webservers use the PDO class in PHP to establish connections with the database and execute queries. The database design for ViNeL is presented in the Appendix as figure 5.

VM Hypervisor
The server has the most current version of Oracle VM VirtualBox installed (as of writing: 4.3.22). The server was also configured to run phpVirtualBox to provide a web-based GUI for easier VM management. The VirtualBox user (vinel) is the same user running the Apache HTTP server. A dedicated folder on the NAS server was

created to be the default path for VirtualBox to install virtual machines and store VM images.

Application Features

PHP was used as the primary programming language of the ViNeL application. The following features were implemented using PHP scripts:

1. User Access Control – Each user has an account in ViNeL. Users are only given access to VMs that they have created. Users are grouped into roles and access is restricted by group membership.

2. System Information Dashboard – Upon logging in, users can view the current percent of storage, memory, and CPU in use. If left open, the dashboard refreshes every minute. The dashboard is shown in Figure 6 in the appendix.

3. Launch Instances and Networks – By taking advantages of the Linked Clone (Oracle Corporation, 2004-2015) feature in VirtualBox, users can a launch instances or networks depending upon their user role. Users must specify a reservation length for how long they would like to have the VM instance available for use.

4. View Personal Instances and Networks – Users can view instances and networks that they have created and chose to end their reservations early if they are finished using their created instance

5. Add and Delete Users – Elevated users have the ability to upload a CSV file of users to create user accounts. Those accounts can also be removed via the web interface.

6. Import Images and Networks – Elevated users can upload and import images and networks for use by all other members.

7. Delete Images and Networks – Elevated users may delete images and networks if they will never be used again or if there is an error with the configuration that cannot be corrected.

8. Enable and Disable Images - Elevated users may disable images without deleting the files in order to prevent users from accesses the VM. Images may be re-enabled at any time.

Virtual Machine Image Creation

Users are given the option of creating virtual machines on a local machine or in the server using the phpVirtualBox interface. Image creation follows the typical setup process described earlier in the paper. After the image has been configured, it can be uploaded to the server. The server will automatically import the image into VirtualBox and made it available for use. For network images, system users can supply network names and the system will create an internal DHCP server for a network containing only those virtual machines.

At the initial deployment of ViNeL, a virtual machine image was created for each of the operating systems listed in table 3, contained in the appendix.

These images were created based on operating systems used in the 2014 SECCDQC (SECCDC, 2014), 2015 SECCDC Qualification (SECCDC, 2015), and the 2015 SECCDC Regional (SECCDC, 2015) competitions.

For single instances (lab use), each image was configured to have all updates installed. Network instances (competition use) are not configured with updates. Once the configuration is complete, the image was uploaded via the application website, which saved it to the NAS and made it available for users to deploy.

5. RESULTS

A virtual networking lab was successfully created that met all but one the requirements of the CYBER DEFENSE CLUB. While the ViNeL application is a self-service, platform-independent, free networking working lab, the system cannot currently support 15 lab users at a time. The server hardware would need to be updated or an additional server added to accommodate more users. The system as configured can support 10 simultaneous users with reasonable performance. The competition practice use portion of the application is fully functional and can support all team members practicing at the same time.

6. DISCUSSION

While the implementation of the ViNeL application has provided a solution to the CYBER DEFENSE CLUB problem of the distribution of system configurations, this project does have broader applications. The application can be adapted for academic purposes where the curriculum requires the use of atypical setups and software configurations. Student could use instance of pre-configured machines to complete training on a variety of software product or jump directly into learning new operating systems without the tedious installation process.

In industry, system administrators can use a similar setup to test the compatibility of new software and updates to be deploy across enterprise systems. Software developers can also use the networking lab to test their product across multiple platforms easily and efficiently. While many businesses have similar setups, the one described here can be implemented at very low cost.

7. CONCLUSIONS

The ViNeL application has opened up many possibilities for the Cyber Defense Club. The leaders of the club can now devote more time to research and delve further into security implementations instead of hours of computer configurations. The fact that ViNeL is built on open source product and freely available licenses through DreamSpark will give peace of mind to CYBER DEFENSE CLUB leaders, as there was no need to transfer license information from year to year, nor will money need to be set aside for license purchases each year.

ViNeL can also be used by faculty to advance the level of development and experimentation seen in courses. It can level the computer playing field by insuring each student has the ability to access numerous servers and applications regardless of the physical limitation of their personal computers.

In the future, the following adjustments can be made to the server to provide a better software for users:
1. <u>Save Images</u> – Users can download the current configuration of their VM when their reservation is complete or add the configuration to the list of available images.
2. <u>Custom Launch Scripts</u> – Users can input Bash or Batch scripts to run in the VM during boot to custom their instance from the current configuration.
3. <u>Automated VirtualBox Updates</u> – The system will automatically update VirtualBox Guest Additions on each VM image when a new update becomes available.
4. <u>Detailed Networking</u> – Elevated users have the option to enter more detailed networking information (IP addresses, subnets, etc.) when importing an image.

8. REFERENCES

2015 Southeast Collegiate Cyber Defense Qualification Competition (SECCDQC) Team Packet. (2015, February). Kennesaw.

Amazon EC2. (2014, August 21). Retrieved from Amazon Web Services: http://aws.amazon.com/ec2/

Bootstrap Core Team. (n.d.). *Dashboard Template for Bootstrap*. Retrieved January 31, 2015, from http://getbootstrap.com/examples/dashboard/

Brock, M. (2014, June 23). *Security hardening on Ubuntu Server 14.04*. Retrieved February 16, 2015, from http://blog.mattbrock.co.uk/hardening-the-security-on-ubuntu-server-14-04/

Chart.js | Open source HTML5 Charts for your website. (n.d.). Retrieved February 13, 2015, from http://www.chartjs.org/

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing.* Computer Security Division. Gaithersburg: National Institute of Standards and Technology.

Microsoft DreamSpark Premium Subscription Agreement. (2015, April 1). Retrieved from Microsoft DreamSpark: https://www.dreamspark.com/Institution/DSP-EULA.aspx

Oracle Corporation. (2004-2015). *Oracle VM VirtualBox User Manual.* Oracle Corporation.

Oracle VM Virtualbox. (2015, January 17). *Oracle VM VirtualBox*. Retrieved from VirtualBox: https://www.virtualbox.org/

SECCDC. (2014, February 15). *2014 SECCDQC Team Packet.* Retrieved from Southeast Collegiate Cyber Defense Competition: http://www.seccdc.org/2014_docs/2014SECCDCQTeamPack_Final.pdf

SECCDC. (2015, March 30). 2015 Southeast Collegiate Cyber Defense Competition (SECCDC) Team Packet. Kennesaw.

The Apache Software Foundation. (2015, January 17). *Apache VCL*. Retrieved from Apache VCL: https://vcl.apache.org/

Virtual Machines. (2015, January 21). Retrieved from Microsoft Azure: http://azure.microsoft.com/en-us/services/virtual-machines/

Appendices

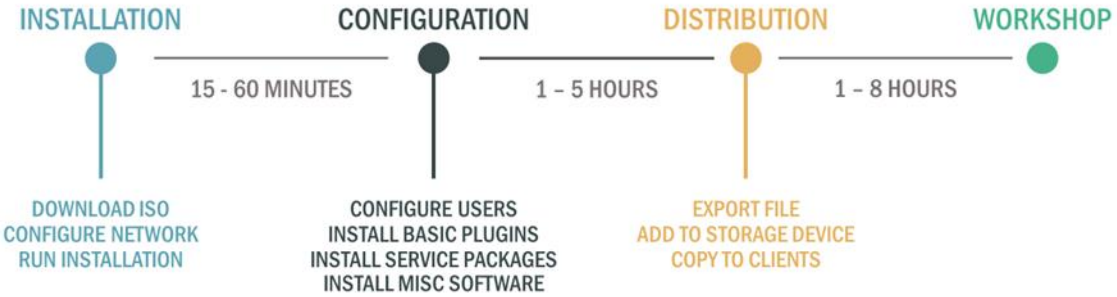


Figure 1 - Lab Setup Time

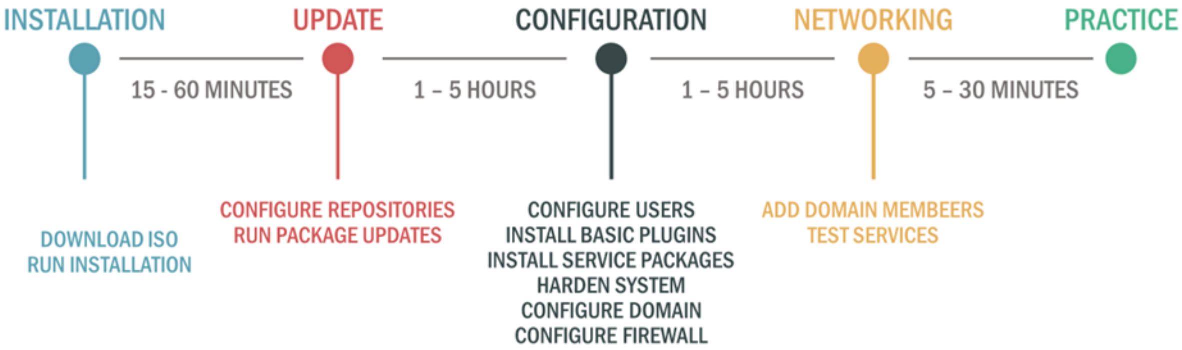


Figure 2 - Competition Practice Setup Timeline

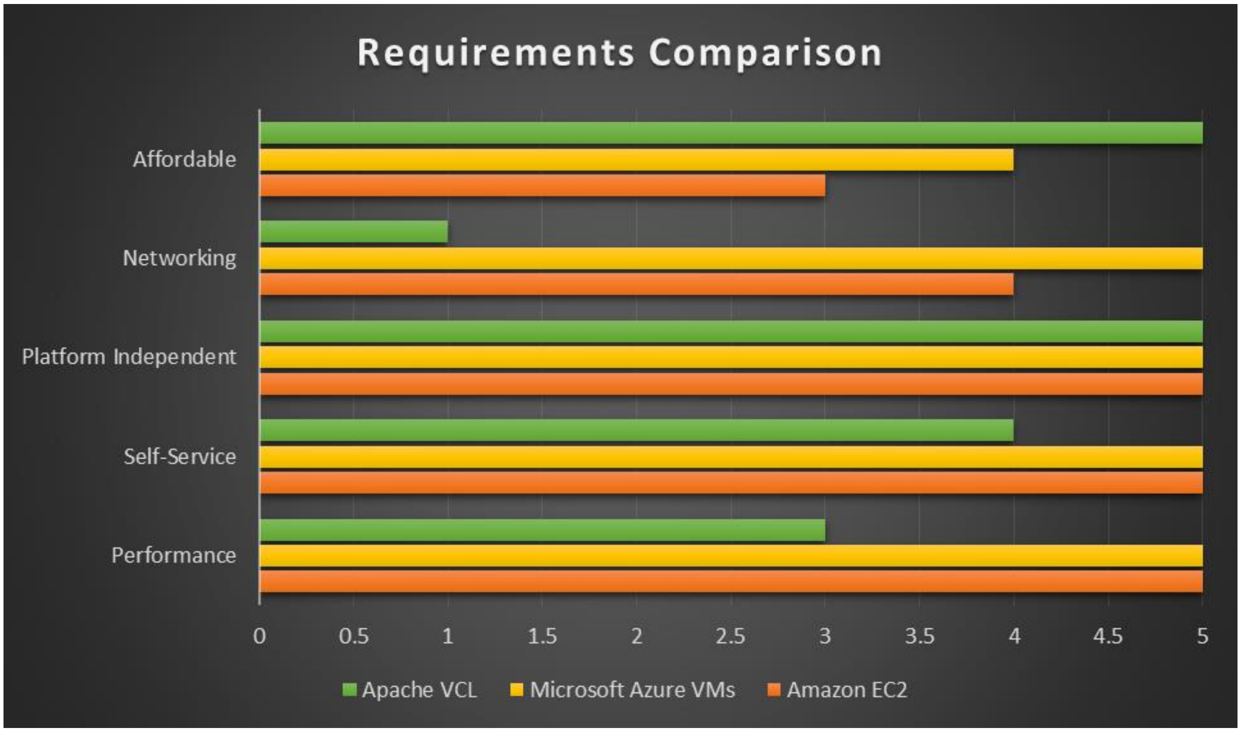


Figure 3 - Requirements Comparison Chart

	Amazon EC2			Microsoft Azure VMs		
	Lab	Competition		Lab	Competition	
Instance Type	t2.small	t2.small	t2.medium	A1	A1	A2
CPU	1	1	2	1	1	2
RAM (GB)	2	2	4	1.75	1.75	3.5
Storage (GB)	15	15	15	40	40	60
Price-per-hour	$0.03	$0.03	$0.05	$0.08	$0.08	$0.15
Hours	4	12	12	4	12	12
Quantity	20	6	1	20	6	1
Additional Charges	$1.50	$9.00	$1.50	N/A	N/A	N/A
Monthly Cost	$3.58	$10.87	$2.12	$6.16	$5.54	$1.82
Total	$16.57			$13.52		

Table 1 - Cloud Computing Estimated Monthly Pricing

Feature	Description
Processor	Intel(R) Xeon(R) E5440 @ 2.83GHz
CPU Cores	8
Memory	32 GB (8 x 4 GB)
Ethernet Ports	4 ports, 10/100/1000 Mbps
Internal Storage	120 GB SATA
Removable Media	1 CD/DVD+RW

Table 2 - ViNeL Hardware Specifications

Operating System	Interface Type
Debian 6.0.10	Graphical
Debian 8.0.0	Graphical
Ubuntu Server 8.04.4 LTS	Command Line
Ubuntu Desktop 12.04.5 LTS	Graphical
Ubuntu Server 14.04.2 LTS	Command Line
Metasploitable 2.0.0	Command Line
Kali Linux 1.1.0a	Graphical
pfSense 2.0.1	Command Line
pfSense 2.2	Command Line
Gentoo 11.2	Graphical
OpenSUSE 11.3	Graphical
CentOS 3.9	Command Line
CentOS 6.6	Graphical
VMWare ESXi 5.5	Command Line
Windows XP Professional with SP3	Graphical
Windows Server 2003 R2 Enterprise with SP2	Graphical
Windows Server 2008 Enterprise with SP2	Graphical
Windows Server 2008 R2 Standard with SP1	Graphical
Windows 7 Professional with SP1	Graphical
Windows Server 2012 Standard	Command Line
Windows Server 2012 Standard	Graphical
Windows 10 Technical Preview	Graphical

Table 3 - Virtual Machine Operating Systems

Figure 4 – ViNeL Architectural

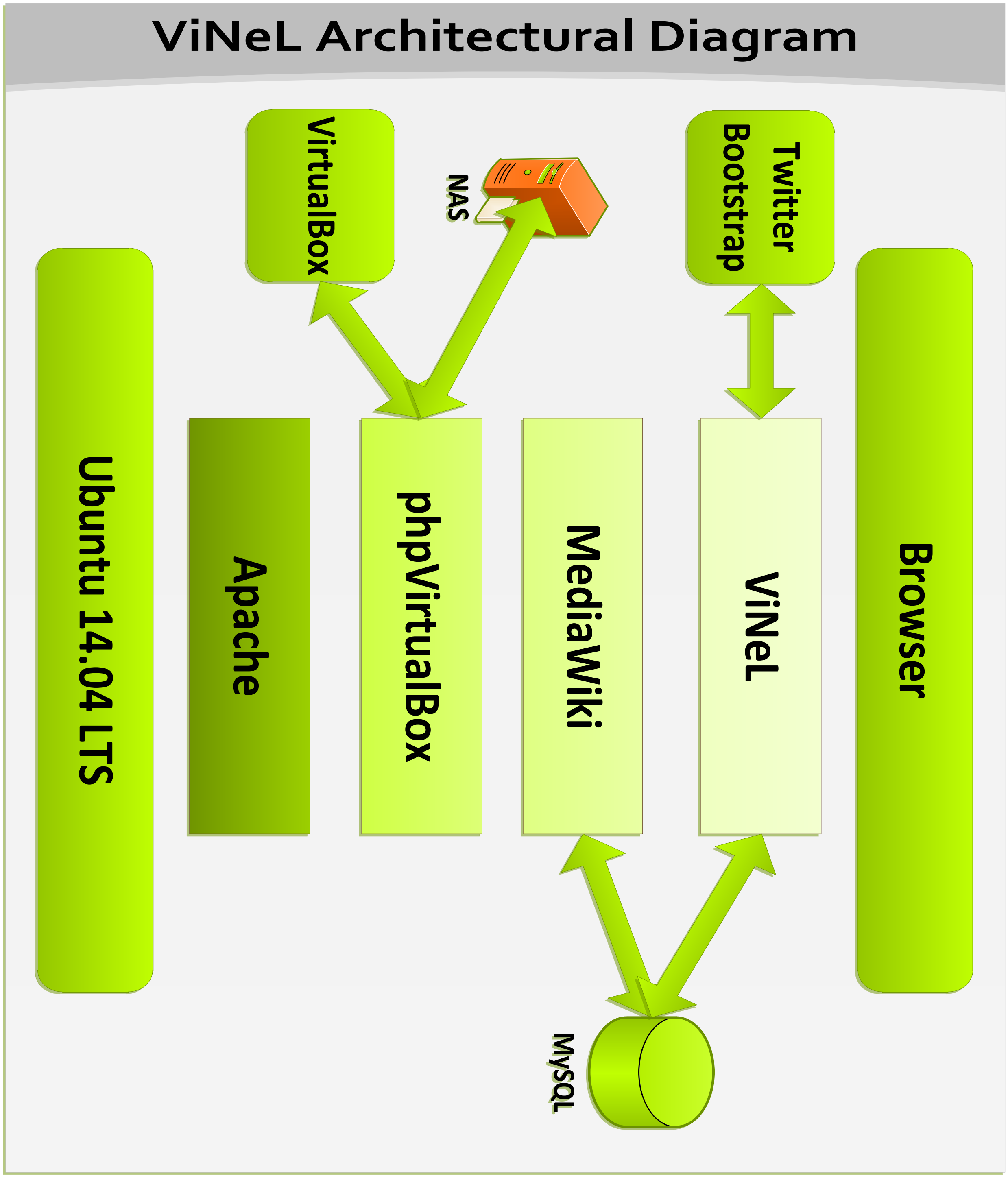


Figure 5 – ViNeL Database

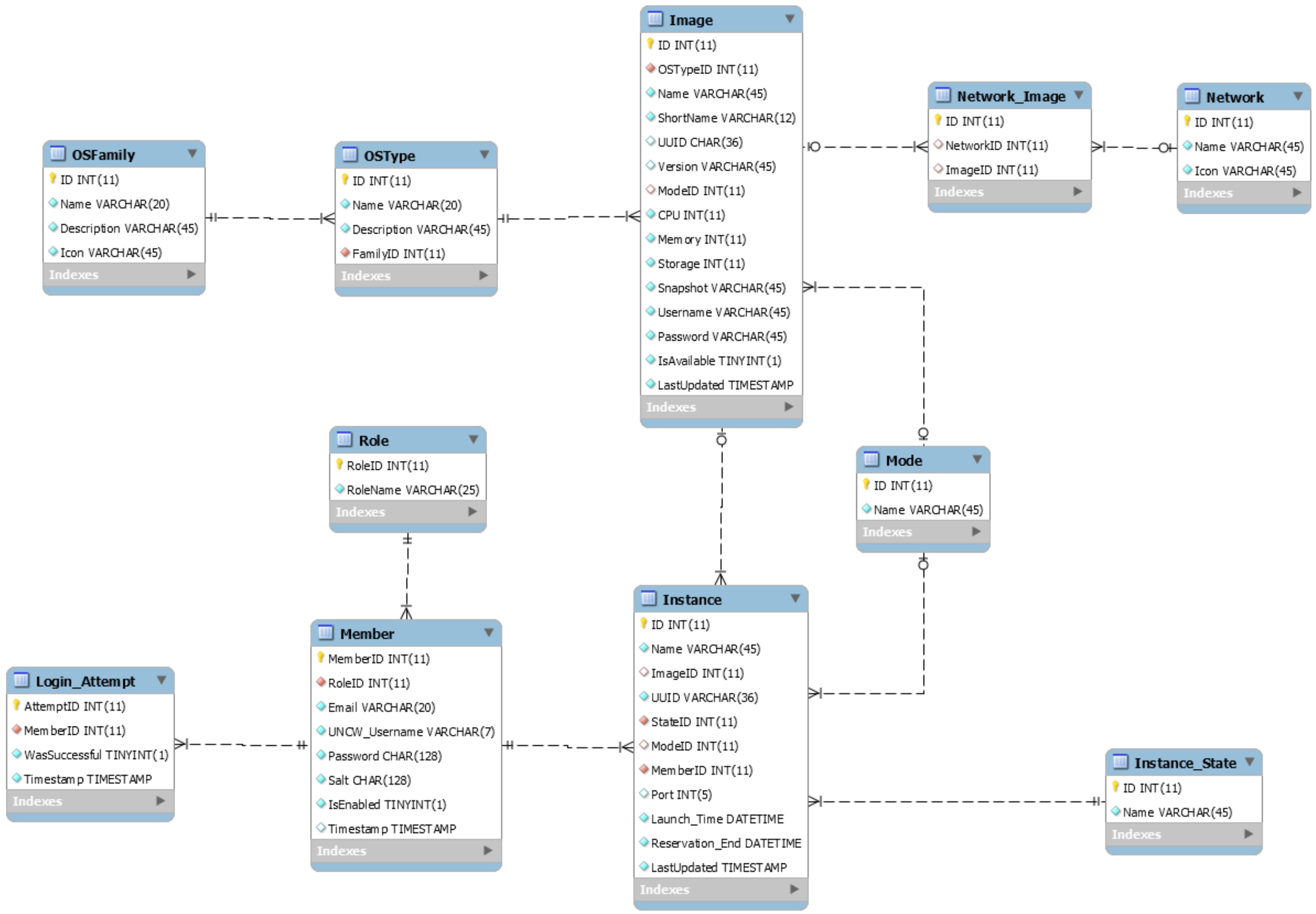
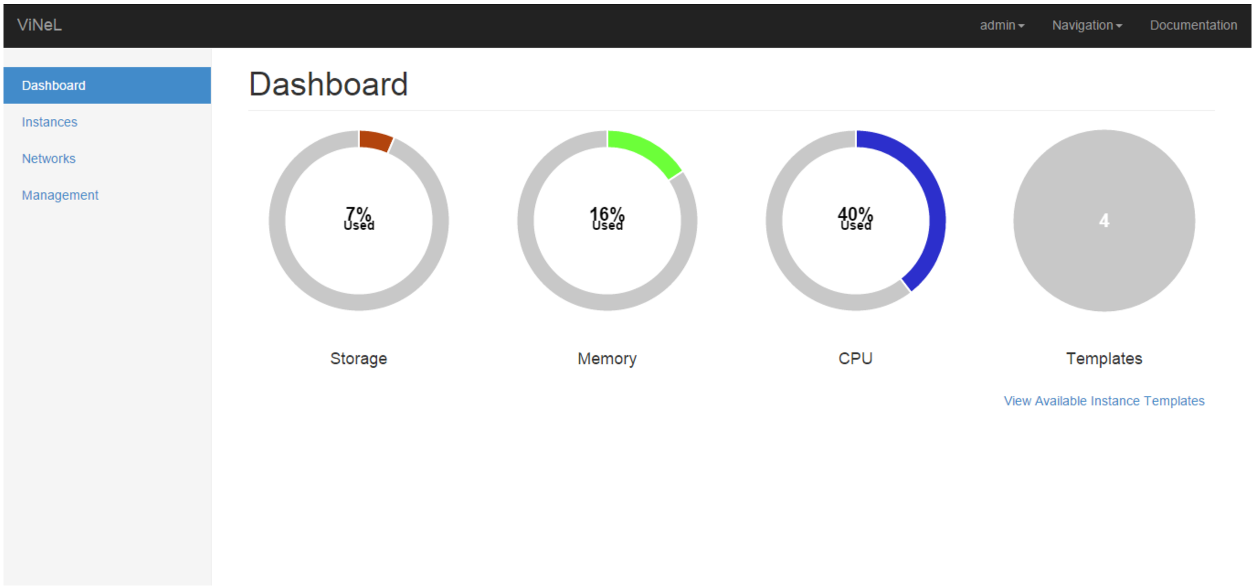


Figure 6 - ViNeL Dashboard


ViNeL
admin
Navigation
Documentation
Dashboard
Instances
Networks
Management
Dashboard
7% Used
16% Used
40% Used
4
Storage
Memory
CPU
Templates
View Available Instance Templates